A Symmetry Preserving Algorithm for Matrix Scaling

Philip A. Knight, Daniel Ruiz, Bora Uçar

A Symmetry Preserving Algorithm for Matrix Scaling

Philip A. Knight, Daniel Ruiz, Bora Uçar

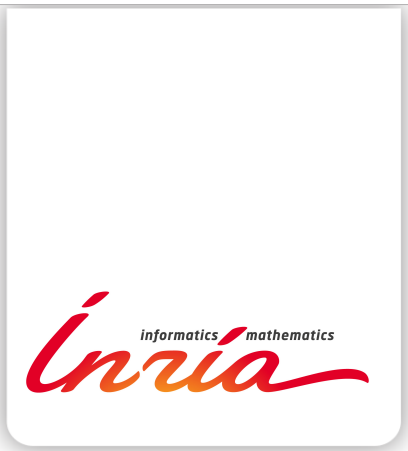
RESEARCH REPORT

N° 7552

November 2012

Project-Team ROMA

ISSN 0249-6399 ISRN INRIA/RR--7552--FR+ENG



A Symmetry Preserving Algorithm for Matrix Scaling

Philip A. Knight, Daniel Ruiz, Bora Uçar

Project-Team ROMA

Research Report n° 7552 — November 2012 — 33 pages

Abstract: We present an iterative algorithm which asymptotically scales the ∞-norm of each row and each column of a matrix to one. This scaling algorithm preserves symmetry of the original matrix and shows fast linear convergence with an asymptotic rate of 1/2. We discuss extensions of the algorithm to the one-norm, and by inference to other norms. For the 1-norm case, we show again that convergence is linear, with the rate dependent on the spectrum of the scaled matrix. We demonstrate experimentally that the scaling algorithm improves the conditioning of the matrix and that it helps direct solvers by reducing the need for pivoting. In particular, for symmetric matrices the theoretical and experimental results highlight the potential of the proposed algorithm over existing alternatives.

Key-words: Sparse matrices, matrix scaling, equilibration

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Un algorithme pour mettre des matrices à l'échelle tout en préservant la symétrie

Résumé : Nous décrivons un algorithme itératif qui, asymptotiquement, met une matrice à l'échelle de telle sorte que chaque ligne et chaque colonne est de taille 1 dans la norme infini. Cet algorithme préserve la symétrie. De plus, il converge assez rapidement avec un taux asymptotique de 1/2. Nous discutons la généralisation de l'algorithme à la norme 1 et, par inférence, à d'autres normes. Pour le cas de la norme 1, nous établissons que l'algorithme converge avec un taux linéaire. Nous démontrons expérimentalement que notre algorithme améliore le conditionnement de la matrice et qu'il aide les méthodes directes de résolution en réduisant le pivotage. Particulièrement pour des matrices symétriques, nos résultats théoriques et que expérimentaux mettent en valeur l'intérêt de notre algorithme par rapport aux algorithmes existants.

Mots-clés : Matrices creuses, mise à l'échelle, factorisation des matrices creuses.

1 Introduction

Scaling a matrix consists of pre- and post-multiplying the original matrix by two diagonal matrices. We consider the following scaling problem: given a large, sparse matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, find two positive diagonal matrices $\mathbf{D}$ and $\mathbf{E}$ such that all rows and columns of the scaled matrix $\widehat{\mathbf{A}} = \mathbf{DAE}$ have the same length in the ∞-norm. We propose an iterative algorithm for this purpose, which has the desirable feature to preserve symmetry when it holds, and we investigate the convergence properties of the proposed algorithm. We discuss the extension of the algorithm to the 1-norm, and describe how it can be used for other norms as well.

Scaling or equilibration of data in linear systems of equations of the form $\mathbf{Ax} = \mathbf{b}$ is a topic of great importance. In this setting, if $\widehat{\mathbf{A}}$ denotes the scaled matrix $\widehat{\mathbf{A}} = \mathbf{DAE}$, then one solves the equation $\widehat{\mathbf{A}}\hat{\mathbf{x}} = \hat{\mathbf{b}}$, where $\hat{\mathbf{x}} = \mathbf{E}^{-1}\mathbf{x}$ and $\hat{\mathbf{b}} = \mathbf{Db}$, with the expectation that the scaled system has a better conditioning and hence the results are more accurate. Assuming a scaled system also becomes helpful in designing algorithms for linear systems. For example, Bunch and Parlett [13] prove the stability of their factorization method by assuming that the rows (hence the columns) of the symmetric input matrix have ∞-norm equal to 1.

There are several well-known algorithms (see for example [16, Section 4.12], [20, Section 3.5.2] and [28]) for scaling matrices. The well-known row and column scaling methods are the most classical scaling methods. In the row scaling method, each row in the original matrix is divided by its norm. Different norms, such as the ∞-norm or the 1-norm, may be considered, depending on the application. The column scaling is identical to the row scaling, except that it works on the columns of the original matrix. A more elaborate scaling method is proposed by Curtis and Reid [14] and is implemented as `MC29` in the HSL Mathematical Software Library (`http://www.hsl.rl.ac.uk/`). This method aims to make the nonzeros of the scaled matrix close to one by minimizing the sum of the squares of the logarithms of the moduli of the nonzeros. `MC29` reduces that sum in a global sense and therefore should be useful on a wide range of sparse matrices. The routine `MC30` in the HSL library is a variant of `MC29` for symmetric matrices. Scaling can also be combined with permutations (see [18] and the HSL routine `MC64`). The matrix is first permuted so that the product of the absolute values of entries on the diagonal of the permuted matrix is maximized. Then the matrix is scaled so that the diagonal entries are one and the off-diagonals are less than or equal to one. This has been shown [19] to be useful for finding a good sequence of pivots for sparse direct solvers and for building good incomplete LU preconditioners for iterative methods.

In the 1960s, some optimality properties in terms of condition numbers for scaled matrices with all rows or all columns of norm of 1 were shown [5, 6, 34]. In particular, the optimal scaling of a matrix $\mathbf{A}$ which minimizes the condition number in the ∞-norm is characterized by both $\mathbf{DAE}$ and $\mathbf{E}^{-1}\mathbf{A}^{-1}\mathbf{D}^{-1}$ having equal row sums of absolute values [5]. Other optimality results for one sided scaling, i.e., $\mathbf{DA}$ or $\mathbf{AE}$ are also shown [34], again based on the equivalence of the norms of rows or columns in the ∞- and 1-norms.

The organization of the paper is as follows. The proposed algorithm is introduced in detail in Section 2. The convergence towards the stationary state mentioned above is at least linear, with an asymptotic rate of convergence of $\frac{1}{2}$,

and this is clearly demonstrated in the same section. We also indicate some particular properties that this algorithm yields for the scaled matrices. In Section 3, we extend the algorithm to the case of other norms. Following the discussion in [24], we establish under which hypothesis the algorithm is also convergent in the case of the 1-norm, and we comment on the generalization of these results with respect to what was stated in [24]. We present numerical results in Section 4. Concluding remarks are given in Section 5.

2 The algorithm

Consider a general $m \times n$ real matrix $\mathbf{A}$. For $i = 1, \ldots, m$, let $\mathbf{r}_i = \mathbf{a}_{i*}^T \in \mathbb{R}^{n\times 1}$ denote the ith row of $\mathbf{A}$ as a vector, and for $j = 1, \ldots, n$, let $\mathbf{c}_j = \mathbf{a}_{*j} \in \mathbb{R}^{n\times 1}$ denote the jth column of $\mathbf{A}$. Furthermore, let $\mathbf{R}$ and $\mathbf{C}$ denote the $m \times m$ and $n \times n$ diagonal matrices given by:

$$\mathbf{R} = \operatorname{diag}\left(\sqrt{\|\mathbf{r}_i\|_\infty}\right)_{i=1,\ldots,m} \quad \text{and} \quad \mathbf{C} = \operatorname{diag}\left(\sqrt{\|\mathbf{c}_j\|_\infty}\right)_{j=1,\ldots,n} \tag{1}$$

where $\|\cdot\|_\infty$ stands for the ∞-norm of a real vector (that is the maximum entry in absolute value; sometimes called the max norm). If a row (or a column) in $\mathbf{A}$ has all entries equal to zero, we replace the diagonal entry in $\mathbf{R}$ (or $\mathbf{C}$ respectively) by 1. In the following, we will assume that this does not happen; zero rows or columns should be taken away to reduce the linear system.

One can scale the matrix $\mathbf{A}$ on both sides, forming the scaled matrix $\widehat{\mathbf{A}}$ in the following way

$$\widehat{\mathbf{A}} = \mathbf{R}^{-1}\mathbf{A}\mathbf{C}^{-1} \,. \tag{2}$$

1: $\mathbf{A}^{(0)} \leftarrow \mathbf{A}$
2: $\mathbf{D}^{(0)} \leftarrow \mathbf{I}_m$
3: $\mathbf{E}^{(0)} \leftarrow \mathbf{I}_n$
4: **for** $k = 0, 1, 2, \ldots$ **until** convergence **do**
5: $\quad \mathbf{R} \leftarrow \operatorname{diag}\left(\sqrt{\|\mathbf{r}_i^{(k)}\|_\infty}\right)_{i=1,\ldots,m}$ ▶$\mathbf{r}_i^{(k)}$ is the ith row of $\mathbf{A}^{(k)}$
6: $\quad \mathbf{C} \leftarrow \operatorname{diag}\left(\sqrt{\|\mathbf{c}_j^{(k)}\|_\infty}\right)_{j=1,\ldots,n}$ ▶$\mathbf{c}_j^{(k)}$ is the jth column of $\mathbf{A}^{(k)}$
7: $\quad \mathbf{A}^{(k+1)} \leftarrow \mathbf{R}^{-1}\mathbf{A}^{(k)}\mathbf{C}^{-1}$
8: $\quad \mathbf{D}^{(k+1)} \leftarrow \mathbf{D}^{(k)}\,\mathbf{R}^{-1}$
9: $\quad \mathbf{E}^{(k+1)} \leftarrow \mathbf{E}^{(k)}\,\mathbf{C}^{-1}$

Algorithm 1: Simultaneous row and column scaling in the ∞-norm

The idea of the algorithm we propose is to iterate on that process, resulting in Algorithm 1. The convergence is obtained when

$$\max_{1\le i\le m}\left\{\left|1 - \|\mathbf{r}_i^{(k)}\|_\infty\right|\right\} \le \varepsilon \quad \text{and} \quad \max_{1\le j\le n}\left\{\left|1 - \|\mathbf{c}_j^{(k)}\|_\infty\right|\right\} \le \varepsilon \tag{3}$$

for a given value of $\varepsilon > 0$.

We note that in an actual implementation one does not need to store the iterates $\mathbf{A}^{(k)}$, rather one can access it through left and right multiplications, respectively, with the current scaling matrices $\mathbf{D}^{(k)}$ and $\mathbf{E}^{(k)}$.

2.1 A salient property

We highlight that the proposed iterative scaling procedure preserves the symmetry in the original matrix. In fact, this is one of our main motivations. If the given matrix $\mathbf{A}$ is symmetric, then the diagonal matrices $\mathbf{R}$ and $\mathbf{C}$ in (1) are equal and, consequently, matrix $\widehat{\mathbf{A}}$ in (2) is symmetric, as is the case for the matrices $\mathbf{A}^{(k)}$ at any iteration in Algorithm 1. This is not the case for most scaling algorithms which alternately scale rows followed by columns or vice-versa.

In the case of unsymmetric matrices, one may consider the use of the Sinkhorn–Knopp iterations given in [31] with the ∞-norm in place of the 1-norm. This method simply normalizes all rows and then all columns of $\mathbf{A}$, and iterates on this process until convergence. In ∞-norm, the convergence is achieved after one single step. Because of its simplicity, this method is very appealing. Notice, however, that the Sinkhorn–Knopp iteration may provide very different results when applied to $\mathbf{A}$ or to $\mathbf{A}^T$. As opposed to that, and this is linked to the first comment above, Algorithm 1 does provide exactly the same results when applied to $\mathbf{A}$ or $\mathbf{A}^T$ in the sense that the scaled matrix obtained on $\mathbf{A}^T$ is the transpose of that obtained on $\mathbf{A}$. We have quoted the Sinkhorn–Knopp method [31] in particular because it has been originally proposed by the authors to obtain *doubly stochastic* matrices (that is nonnegative matrices with all rows and columns of 1-norm equal to one), and we shall come back to this issue with respect to Algorithm 1 later.

2.2 Convergence rate

The particular case when the matrix $\mathbf{A}$ has all its rows and columns with ∞-norm equal to one is clearly a fixed point for the iterations in Algorithm 1. Also, if $\mathbf{A}$ is a square matrix in which the absolute value of each diagonal element (after a permutation of columns) is greater than or equal to the absolute value of any other entry in the corresponding row and column, then it can easily be seen that the algorithm converges in one iteration, with a resulting scaled matrix $\mathbf{A}^{(1)}$ with all ones on the diagonal (after the same column permutation).

Concerning the rate of convergence of Algorithm 1 in the more general case, we shall now verify that the algorithm converges in all cases towards the above mentioned stationary point with an asymptotic linear rate of $\frac{1}{2}$.

The first point in the demonstration is to notice that, after the first iteration of the algorithm, all the entries in $\mathbf{A}^{(1)}$ are less than or equal to one in absolute value. This is very easy to see, since all entries a_{ij} in $\mathbf{A}$ are divided by the square roots of two numbers, $\|\mathbf{r}_i^{(k)}\|_\infty$ and $\|\mathbf{c}_j^{(k)}\|_\infty$ respectively, each one of them being greater than or equal to $|a_{ij}|$ itself.

Then, for any subsequent iteration ($k \geq 1$), consider the ∞-norm of any row $\mathbf{r}_i^{(k)}$ or column $\mathbf{c}_j^{(k)}$, and let indices ℓ and p satisfy the equalities $|a_{ip}^{(k)}| = \|\mathbf{r}_i^{(k)}\|_\infty$ and $|a_{\ell j}^{(k)}| = \|\mathbf{c}_j^{(k)}\|_\infty$. With these notations, we can easily verify that both entries $a_{ip}^{(k+1)}$ and $a_{\ell j}^{(k+1)}$ in the scaled matrix $\mathbf{A}^{(k+1)}$ are greater, in absolute value, than the square root of the corresponding value at iteration k, and are still less than one. Indeed, we can write

$$1 \geq |a_{ip}^{(k+1)}| = \frac{|a_{ip}^{(k)}|}{\sqrt{\|\mathbf{r}_i^{(k)}\|_\infty}\sqrt{\|\mathbf{c}_p^{(k)}\|_\infty}} = \frac{\sqrt{|a_{ip}^{(k)}|}}{\sqrt{\|\mathbf{c}_p^{(k)}\|_\infty}} \geq \sqrt{|a_{ip}^{(k)}|}$$

since $|a_{ip}^{(k)}| = \|\mathbf{r}_i^{(k)}\|_\infty$ and $\|\mathbf{c}_p^{(k)}\|_\infty \leq 1$ for any $k \geq 1$. A similar short demonstration enables us to show that

$$\sqrt{|a_{\ell j}^{(k)}|} \leq |a_{\ell j}^{(k+1)}| \leq 1 \, ,$$

for any $k \geq 1$. From this, we can finally write that the iterations in Algorithm 1 provide scaled matrices $\mathbf{A}^{(k)}$, $k = 1, 2, \ldots$ with the following properties

$$\forall k \geq 1, \; 1 \leq i \leq m, \;\; \sqrt{\|\mathbf{r}_i^{(k)}\|_\infty} \leq |a_{ip}^{(k+1)}| \leq \|\mathbf{r}_i^{(k+1)}\|_\infty \leq 1 \, , \tag{4}$$

and

$$\forall k \geq 1, \; 1 \leq j \leq n, \;\; \sqrt{\|\mathbf{c}_j^{(k)}\|_\infty} \leq |a_{\ell j}^{(k+1)}| \leq \|\mathbf{c}_j^{(k+1)}\|_\infty \leq 1 \, , \tag{5}$$

which shows that both row and column norms must converge to 1. To conclude our demonstration, we just need to see that

$$1 - \|\mathbf{r}_i^{(k+1)}\|_\infty = \frac{1 - \|\mathbf{r}_i^{(k+1)}\|_\infty^2}{1 + \|\mathbf{r}_i^{(k+1)}\|_\infty} \leq \frac{1 - \|\mathbf{r}_i^{(k)}\|_\infty}{1 + \|\mathbf{r}_i^{(k+1)}\|_\infty},$$

and that similar equations hold for the columns as well. This completes the proof of the linear convergence of Algorithm 1 with an asymptotic rate of $\frac{1}{2}$.

A small example, taken from the discussion in [27], actually shows that this asymptotic rate is sharp. To illustrate that, let us consider the following 2×2 matrix with a badly scaled row

$$\mathbf{A} = \begin{pmatrix} \alpha & \alpha \\ 1 & 1 \end{pmatrix}.$$

If $\alpha \ll 1$, then iteration k ($k \geq 1$) of the algorithm provides the following matrices:

$$\mathbf{D}^{(k)} = \begin{pmatrix} \alpha^{-(1-\frac{1}{2^k})} & 0 \\ 0 & 1 \end{pmatrix}, \; \mathbf{A}^{(k)} = \begin{pmatrix} \alpha^{\frac{1}{2^k}} & \alpha^{\frac{1}{2^k}} \\ 1 & 1 \end{pmatrix}, \; \mathbf{E}^{(k)} = \mathbf{I}_2,$$

converging to the situation where the scaled matrix $\widehat{\mathbf{A}}$ is the matrix with all ones, $\mathbf{D}$ has its first diagonal entry equal to α^{-1}, and $\mathbf{E}$ stays as the identity matrix. The above example shows that we cannot expect in general (apart from some particular cases) to prove faster convergence for Algorithm 1 than the linear rate of $\frac{1}{2}$.

2.3 Comparison with Bunch's algorithm

As we have stated before, Algorithm 1 is well suited for symmetric scaling of symmetric matrices. For the ∞-norm case, Bunch [12] also developed an efficient symmetric scaling algorithm. Bunch's algorithm processes the rows (of the lower triangular part of the given matrix) sequentially in such a way that the largest entry seen in each row is made to be ± 1 in the scaled matrix. After this processing of the rows, the scaling found for the ith row is applied to the ith column of the whole matrix. The sequential nature of the algorithm renders it sensitive to symmetric permutations applied to the original matrix. For example, any diagonal nonzero can be symmetrically permuted to the first

position so that that entry is one in the scaled matrix. On the other hand, it is easy to see that the proposed algorithm (Algorithm 1) is independent of any permutations (not even necessarily symmetric) applied to the original matrix in the sense that the scaling matrices computed for the permuted matrix would be equal to the permuted scaling matrices computed for the original matrix. Bunch's algorithm runs in $\mathcal{O}(\text{nnz})$-time, equivalent to one iteration of the proposed algorithm. However, Algorithm 1 is amenable to parallelism (as we have shown in [2]), whereas Bunch's algorithm is intrinsically sequential.

3 Extensions to other norms

A natural idea would be to change the norm used in Algorithm 1, and to try for instance the 2-norm or the 1-norm because of the optimal properties they induce (see [24, 34]), and still expect convergence towards an equilibrated situation with all rows and columns of length 1 in the corresponding norm. We shall see, in the remainder of this section, that this will usually, but not always, work and we investigate the potential and limitations of such extensions. As opposed to the case of the infinity norm, where different algorithms can raise very different solutions, scalings in the 1-norm or the 2-norm, when they exist, can be considered as unique in the sense that the scaled matrix $\widehat{\mathbf{A}}$ is essentially unique.

With respect to the extension of Algorithm 1 to the scaling of matrices in other norms, the case of the 1-norm is central. Indeed, Rothblum et al. have shown [26, page 13] that the problem of scaling a matrix $\mathbf{A}$ in the ℓ_p-norm, for $1 < p < \infty$ can be reduced to the problem of scaling the pth *Hadamard power of* $\mathbf{A}$, i.e., the matrix $\mathbf{A}^{[p]} = [a_{ij}^p]$, in the 1-norm. We can apply that discussion to Algorithm 1 by replacing the matrix $\mathbf{A}$ with $\mathbf{A}^{[p]}$ and then by taking the *Hadamard* pth root, e.g., $\mathbf{D}_1^{[1/p]} = [d_{ii}^{1/p}]$, of the resulting iterates. For this reason, we shall analyse, in the following of this section, the convergence properties of Algorithm 1 for the 1-norm only, knowing that these will implicitly drive the conclusions for any of the ℓ_p norms, for $1 < p < \infty$.

3.1 Background

The idea of equilibrating a matrix such that the 1-norm of the rows and columns are all 1 is not new, and has been the subject of constant efforts since the 1960's, and even before. Here, we briefly review some of the the previous work, and give convergence proof of Algorithm 1 for the 1-norm.

Sinkhorn and Knopp [31] studied a method for scaling square nonnegative matrices to doubly stochastic form, that is a nonnegative matrix with all rows and columns of equal 1-norm. In [29], Sinkhorn originally showed that: *Any positive square matrix of order n is diagonally equivalent to a unique doubly stochastic matrix of order n, and the diagonal matrices which take part in the equivalence are unique up to scalar factors.* Later, a different proof for the existence part of Sinkhorn's theorem with some elementary geometric interpretations was given [9].

This result was further extended to the case of nonnegative, nonzero matrices [31]. A few definitions are necessary to state the result. A square $n \times n$ nonnegative matrix $\mathbf{A} \geq 0$ is said to have *support* if there exists a permutation

σ such that $a_{i,\sigma(i)} > 0$, $1 \leq i \leq n$. Note that matrices not having support are matrices for which no full transversal can be found (see [16, page 107]), i.e., there is no column permutation making the diagonal zero-free, and are thus *structurally singular*. A matrix $\mathbf{A}$ is said to have *total support* if every positive entry in $\mathbf{A}$ can be permuted into a positive diagonal with a column permutation. A nonnegative nonzero square matrix $\mathbf{A}$ of size $n > 1$ is said to be *fully indecomposable* if there does not exist permutation matrices $\mathbf{P}$ and $\mathbf{Q}$ such that $\mathbf{PAQ}$ is of the form

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{A}_{22} \end{pmatrix},$$

with $\mathbf{A}_{11}$ and $\mathbf{A}_{22}$ being square matrices. The term bi-irreducible is also used for fully indecomposable matrices [16, Chapter 6]. Sinkhorn and Knopp [31] established that their scaling algorithm, which simply iterates on normalizing all rows and columns in the matrix $\mathbf{A}$ alternately, converges to a doubly stochastic limit $\widehat{\mathbf{A}}$ if and only if the matrix $\mathbf{A}$ has support. The doubly stochastic limit $\widehat{\mathbf{A}}$ can be represented as $\mathbf{DAE}$ (meaning that $\mathbf{A}$ and $\widehat{\mathbf{A}}$ are diagonally equivalent) if and only if $\mathbf{A}$ has total support and, if the support of $\mathbf{A}$ is not total, then there must be a positive entry in $\widehat{\mathbf{A}}$ converging to 0. Additionally, it has been shown that the diagonal matrices $\mathbf{D}$ and $\mathbf{E}$, when they exist, are unique up to some scalar factors if and only if the matrix $\mathbf{A}$ is fully indecomposable [31]. Brualdi et al. [10] independently showed the same diagonal equivalence between $\mathbf{A}$ and a doubly stochastic matrix when $\mathbf{A}$ is a direct sum of fully indecomposable matrices. It has been shown that a matrix $\mathbf{A}$ has total support if and only if there exist permutation matrices $\mathbf{P}$ and $\mathbf{Q}$ such that $\mathbf{PAQ}$ is a direct sum of fully indecomposable matrices [25, Theorem 1 (ii)].

Different contributions have also been made in the study of convergence of the Sinkhorn–Knopp method under various hypothesis. Geometric convergence rate for positive matrices has been shown in [30]. Geometric convergence of the method for the case of nonnegative matrices with total support has been shown in [33]. In [1], the converse of the above result has been established, i.e., geometric convergence of the Sinkhorn–Knopp algorithm implies total support for a nonnegative matrix. The explicit rates of convergence for fully indecomposable matrices are given in [21].

In [24], Parlett and Landis present three iterative scaling algorithms with experimental evidence of better average and worst-case convergence behavior than the Sinkhorn–Knopp method (in at least one of the three algorithms). They also give a generalized version of the convergence theorem of Sinkhorn and Knopp [31], including a characterization of scaling algorithms that will converge to a doubly stochastic matrix when the starting matrix $\mathbf{A}$ has support. Such algorithms are called *diagonal product increasing* (DPI) algorithms. In Appendix A, we recall and extend partly those results from Parlett and Landis [24, Theorem 1, page 64], and we emphasize the specific generic properties that our algorithm actually fulfills and that are sufficient to ensure convergence of scaling algorithms in the 1-norm in general.

In the following of this section, we first establish that Algorithm 1 respects those generic properties given in Appendix, and that it converges in the one-norm as well as in any other of the p-th norms, $p < \infty$, under some specific conditions for the matrix $\mathbf{A}$. We also analyze the convergence rate of the algorithm in the 1-norm (separately for the symmetric and unsymmetric matrices),

allowing a fuller comparison with other balancing algorithms.

3.2 Convergence analysis

We will follow the approach in [24] to show the convergence of Algorithm 1 in the case of the 1-norm. Recall that Algorithm 1 produces a sequence of matrices diagonally equivalent to the starting matrix $\mathbf{A} = \mathbf{A}^{(0)}$ with the following iterates:

$$\begin{aligned} \mathbf{A}^{(k)} &= \left(a_{ij}^{(k)}\right) = \mathbf{D}^{(k)}\mathbf{A}\mathbf{E}^{(k)}, \quad k = 1, 2, \ldots, \\ \mathbf{D}^{(k)} &= \operatorname{diag}\left(d_1^{(k)}, \ldots, d_n^{(k)}\right), \\ \mathbf{E}^{(k)} &= \operatorname{diag}\left(e_1^{(k)}, \ldots, e_n^{(k)}\right), \end{aligned} \tag{6}$$

where $\mathbf{D}^{(0)} = \mathbf{E}^{(0)} = \mathbf{I}$. For notational convenience let $r_i^{(k)}$, $i = 1, \ldots, n$, and $c_j^{(k)}$, $j = 1, \ldots, n$, denote the 1-norm of rows and columns respectively, thus:

$$\begin{aligned} r_i^{(k)} &= \|\mathbf{r}_i^{(k)}\|_1 = \sum_{j=1}^{n} |a_{ij}^{(k)}|, \\ c_j^{(k)} &= \|\mathbf{c}_j^{(k)}\|_1 = \sum_{i=1}^{n} |a_{ij}^{(k)}|. \end{aligned} \tag{7}$$

We also assume for simplicity that $\mathbf{A} \geq 0$, since scaling $\mathbf{A}$ or $|\mathbf{A}|$ will be the same. Under this simplification, the 1-norm of the rows and columns reduces to the row and column sums respectively, $r_i^{(k)} = \sum_{j=1}^{n} a_{ij}^{(k)}$ and $c_j^{(k)} = \sum_{i=1}^{n} a_{ij}^{(k)}$, and to generalize our results to any matrix, one just needs to extend the definition of a doubly stochastic matrix so that the absolute value of the matrix under consideration is doubly stochastic in the usual sense.

Theorem 1. *Given the sequence (6) of diagonal equivalents for* $\mathbf{A}$*, in which*

$$a_{ij}^{(k+1)} = \frac{a_{ij}^{(k)}}{\sqrt{r_i^{(k)}}\sqrt{c_j^{(k)}}}, \quad 1 \leq i, j \leq n,$$

with $r_i^{(k)}$ *and* $c_j^{(k)}$ *given by (7):*

1. *If* $\mathbf{A}$ *has support, then* $\mathbf{S} = \lim_{k\to\infty} \mathbf{A}^{(k)}$ *exists and is doubly stochastic.*
2. *If* $\mathbf{A}$ *has total support, then both* $\mathbf{D} = \lim_{k\to\infty} \mathbf{D}^{(k)}$ *and* $\mathbf{E} = \lim_{k\to\infty} \mathbf{E}^{(k)}$ *exist and* $\mathbf{S} = \mathbf{DAE}$*.*

Before proving the theorem, we list some basic facts about the iterates produced in Algorithm 1 in the 1-norm. We recall the arithmetic-geometric mean inequality that states that if $x_i \geq 0$ for $i = 1, \ldots, n$ then

$$\prod_{i=1}^{n} x_i \leq \left(\sum_{i=1}^{n} \frac{x_i}{n}\right)^n, \tag{8}$$

with equality holding if and only if $x_1 = x_2 = \cdots = x_n$.

Now, using (8), we can write the following for all k

$$\prod_{i=1}^{n} r_i^{(k+1)} \leq \left(\frac{1}{n} \sum_{i=1}^{n} r_i^{(k+1)} \right)^n = \left(\frac{1}{n} \sum_{1 \leq i,j \leq n} \frac{a_{ij}^{(k)}}{\sqrt{r_i^{(k)}} \sqrt{c_j^{(k)}}} \right)^n ,$$

with the same inequality for $\prod_{j=1}^{n} c_j^{(k+1)}$ since

$$\sum_{i=1}^{n} r_i^{(k+1)} = \sum_{j=1}^{n} c_j^{(k+1)} = \sum_{1 \leq i,j \leq n} a_{ij}^{(k+1)} .$$

Additionally, using the Cauchy-Schwarz inequality on the dot-product of the two n^2-vectors $\mathbf{v} = \left(\sqrt{a_{ij}^{(k)}} / \sqrt{r_i^{(k)}} \right)_{1 \leq i,j \leq n}$ and $\mathbf{w} = \left(\sqrt{a_{ij}^{(k)}} / \sqrt{c_j^{(k)}} \right)_{1 \leq i,j \leq n}$, we can write that

$$\sum_{1 \leq i,j \leq n} \frac{a_{ij}^{(k)}}{\sqrt{r_i^{(k)}} \sqrt{c_j^{(k)}}} \leq \sqrt{\sum_{1 \leq i,j \leq n} \frac{a_{ij}^{(k)}}{r_i^{(k)}}} \sqrt{\sum_{1 \leq i,j \leq n} \frac{a_{ij}^{(k)}}{c_j^{(k)}}} = \sqrt{n} \sqrt{n} ,$$

and thus, for all $k \geq 0$, we have

$$\prod_{i=1}^{n} r_i^{(k+1)} \leq \left(\frac{1}{n} \sum_{i=1}^{n} r_i^{(k+1)} \right)^n \leq 1 \text{ and } \prod_{j=1}^{n} c_j^{(k+1)} \leq \left(\frac{1}{n} \sum_{j=1}^{n} c_j^{(k+1)} \right)^n \leq 1 . \tag{9}$$

Notice also that, after the first iteration, all the entries in matrix $\mathbf{A}^{(k)}$ are less than or equal to 1, since both $r_i^{(k)}$ and $c_j^{(k)}$ are greater than $a_{ij}^{(k)}$ by construction.

Following the ideas developed by Parlett and Landis [24], we shall now establish that Algorithm 1 in the 1-norm is *diagonal product increasing* (DPI). In other words, it fulfills the three main properties (P1*), (P2) and (P3), detailed in Appendix A, so that we can recover the conclusions given in Theorem 4 and Theorem 5. These conclusions extend partly those results in [24, Theorem 1] and imply the convergence of DPI algorithms.

Proof. Property (P1*) requires that both sequences $(\prod_{i=1}^{n} d_i^{(k)})_{k \geq 1}$ and $(\prod_{i=1}^{n} e_i^{(k)})_{k \geq 1}$ be independently monotonically increasing. This is directly induced by (9) and by the fact that the iterates in Algorithm 1 satisfy

$$\frac{\prod_{i=1}^{n} d_i^{(k+1)}}{\prod_{i=1}^{n} d_i^{(k)}} = \frac{1}{\sqrt{\prod_{i=1}^{n} r_i^{(k)}}}$$

(and similarly for the column scaling factors). This naturally implies that the sequence $(s_k)_{k \geq 1}$, where $s_k = \prod_{i=1}^{n} d_i^{(k)} e_i^{(k)}$, $k = 1, 2, \ldots$, is monotonically increasing too, which corresponds to the less restrictive property (P1) introduced by Parlett and Landis (and is also exploited in Theorem 4 in Appendix A, (P1*) being only used in Theorem 5).

Property (P3) simply requires that the sum of all the elements in the iterates $\mathbf{A}^{(k)}$ be bounded above by some constant independent of k. From (9), we directly have $\frac{1}{n}\sum_{i=1}^{n} r_i^{(k+1)} \leq 1$, which implies that

$$\sum_{1\leq i,j\leq n} a_{ij}^{(k+1)} \leq n$$

and provides the required property.

Property (P2) requires to verify that

$$\lim_{k\to\infty} r_i^{(k)} = 1 \text{ and } \lim_{k\to\infty} c_j^{(k)} = 1, \ \ 1 \leq i,j \leq n \ . \tag{10}$$

From the arithmetic-geometric mean equality (8) and from the bounds in (9), we can write

$$\prod_{i=1}^{n} r_i^{(k)} c_i^{(k)} \leq \left\{ \sum_{i=1}^{n} \frac{1}{2n} \left(r_i^{(k)} + c_i^{(k)} \right) \right\}^{2n} \leq 1 \ .$$

From (P1) and (P3) above, and from Lemma 1 in Appendix A, we also have that $\lim_{k\to\infty} s_k = \xi > 0$ exists, and consequently that

$$\lim_{k\to\infty} \frac{s_k}{s_{k+1}} = \lim_{k\to\infty} \prod_{i=1}^{n} \sqrt{r_i^{(k)} c_i^{(k)}} = 1 \ .$$

Therefore, we can conclude that

$$\lim_{k\to\infty} \prod_{i=1}^{n} r_i^{(k)} c_i^{(k)} = 1 \text{ and } \lim_{k\to\infty} \sum_{i=1}^{n} \frac{1}{2n} \left(r_i^{(k)} + c_i^{(k)} \right) = 1 \ . \tag{11}$$

Now, since all the elements in $\mathbf{A}^{(k)}$ are less than 1 after the first iteration, we know that each of the two sequences $(r_i^{(k)})_{k\geq 1}$ and $(c_j^{(k)})_{k\geq 1}$, for all $1 \leq i,j \leq n$, are bounded. Let us introduce the sequence $(\mathbf{v}^{(k)})_{k\geq 1}$ of the $2n$-vectors

$$\mathbf{v}^{(k)} = (r_1^{(k)}, \ldots, r_n^{(k)}, c_1^{(k)}, \ldots, c_n^{(k)}) \ ,$$

which is also bounded in $\mathbb{R}^{2n}$ of finite dimension. Consider then any convergent subsequence $(\mathbf{v}^{(q)})_q$ and let

$$x_i = \lim_{q\to\infty} r_i^{(q)}, \ 1 \leq i \leq n \ ,$$

and

$$y_j = \lim_{q\to\infty} c_j^{(q)}, \ 1 \leq j \leq n \ .$$

From (11), we can write

$$\prod_{i=1}^{n} x_i y_i = \left\{ \sum_{i=1}^{n} \frac{1}{2n} (x_i + y_i) \right\}^{2n} = 1 \ ,$$

and since the arithmetic-geometric mean equality only holds if all the elements are equal, we easily see that $x_1 = \cdots = x_n = 1 = y_1 = \cdots = y_n$. Therefore any

convergent subsequence of the bounded sequence $(\mathbf{v}^{(k)})_{k\geq 1}$ in finite dimensional space must have the same limit (made with all ones), which implies that the sequence $(\mathbf{v}^{(k)})_{k\geq 1}$ is necessarily convergent and that (10) holds.

Now, since our algorithm fulfills properties (P1*), (P2) and (P3), we can recover the various conclusions given in Theorem 4 and Theorem 5 in Appendix A. Additionally, since the computation of the scaling factors in Algorithm 1 is only based on the rows and columns sums in matrix $\mathbf{A}^{(k)}$, at each iteration (k), we easily see that the algorithm is permutation insensitive and does not mix information from totally independent subsets of entries in the matrix $\mathbf{A}$, as required in Corollary 2, and this helps to complete the various conclusions raised in Theorem 1. □

We conclude this subsection with some comments. As already recalled in the course of the previous discussion, Parlett and Landis [24], in their Theorem 1, have raised three main properties characterizing iterative scaling algorithms that would converge to a doubly stochastic matrix when the matrix $\mathbf{A}$ has support. Properties (P2) and (P3) used above are partly weakened with respect to the corresponding ones given in [24], and address precisely the case of our Algorithm 1. Theorem 4 actually recovers the same conclusions as in [24] but with these weakened assumptions, and Theorem 5 provides some extensions. The demonstration of these two theorems follows very closely the discussions and developments made by Parlett and Landis. For completeness, we have put anyway in Appendix A all the material about this generalisation, including the complete and detailed proofs of the various results, as well as some more detailed discussion about these differences with respect to the work in [24].

We have one more specific comment for the symmetric case, considering the fact that Algorithm 1 preserves symmetry.

Corollary 1. *1. If $\mathbf{A}$ is symmetric and has support, then Algorithm 1 in the 1-norm builds a sequence of symmetric scalings of $\mathbf{A}$ converging to a symmetric doubly stochastic limit.*

2. If $\mathbf{A}$ is symmetric and has total support, then $\mathbf{A}$ is symmetrically equivalent to a symmetric doubly stochastic matrix $\mathbf{S}$, and Algorithm 1 in the 1-norm builds a convergent sequence of diagonal matrices $\mathbf{D}^{(k)}$ such that $\mathbf{S} = \lim_{k\to\infty} \mathbf{D}^{(k)}\mathbf{A}\mathbf{D}^{(k)}$.

3.3 Rate of convergence for symmetric matrices

Let $\mathbf{e}$ be a vector of ones (dimension should be clear) and $\mathcal{D}(\mathbf{x}) = \mathrm{diag}(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^n$. Note that $\mathbf{x} = \mathcal{D}(\mathbf{x})\mathbf{e}$ and $\mathcal{D}(\mathcal{D}(\mathbf{x})\mathbf{y}) = \mathcal{D}(\mathbf{x})\mathcal{D}(\mathbf{y}) = \mathcal{D}(\mathbf{y})\mathcal{D}(\mathbf{x})$. When we take the square root of a vector this should be interpreted componentwise. We will also assume that $\mathbf{A} \geq 0$ and that it has no zero rows.

We first treat the case when $\mathbf{A}$ is symmetric. Then the 1-norm version of Algorithm 1 loops the following steps (where $\mathbf{A}^{(0)} = \mathbf{A}$ and $\mathbf{E}^{(0)} = \mathbf{I}$)

$$\mathbf{R} = \mathcal{D}(\mathbf{A}^{(k)}e)^{1/2}, \quad \mathbf{E}^{(k+1)} = \mathbf{R}^{-1}\mathbf{E}^{(k)}, \quad \mathbf{A}^{(k+1)} = \mathbf{R}^{-1}\mathbf{A}^{(k)}\mathbf{R}^{-1}. \tag{12}$$

Note that $\mathbf{A}^{(k)} = \mathbf{E}^{(k)}\mathbf{A}\mathbf{E}^{(k)}$ and we can come up with a more compact form of the algorithm as follows. Let $\mathbf{x}_k$ be the vector such that $\mathcal{D}(\mathbf{x}_k) = \mathbf{E}^{(k)}$. Then

$$\begin{aligned}\mathbf{E}^{(k+1)} &= \mathbf{R}^{-1}\mathbf{E}^{(k)} = \mathcal{D}(\mathbf{E}^{(k)}\mathbf{A}\mathbf{E}^{(k)}e)^{-1/2}\mathbf{E}^{(k)} \\ &= \mathcal{D}(\mathcal{D}(\mathbf{x}_k)\mathbf{A}\mathbf{x}_k)^{-1/2}\mathcal{D}(\mathbf{x}_k) = \mathcal{D}(\mathbf{x}_k)^{-1/2}\mathcal{D}(\mathbf{A}\mathbf{x}_k)^{-1/2}\mathcal{D}(\mathbf{x}_k) \\ &= \mathcal{D}(\mathbf{x}_k)^{1/2}\mathcal{D}(\mathbf{A}\mathbf{x}_k)^{-1/2}.\end{aligned}$$

In other words, we can carry out the iteration simply working with the vectors $\mathbf{x}_k$. Namely, let $\mathbf{x}_0 = \mathbf{e}$ (say) and form

$$\mathbf{x}_{k+1} = \sqrt{\frac{\mathbf{x}_k}{\mathbf{A}\mathbf{x}_k}}, \tag{13}$$

where the division is performed componentwise. If $\mathbf{A}$ is unsymmetric we can replace it with

$$\begin{pmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{pmatrix}$$

and extract the row and column scalings from the top and bottom half of $\mathbf{x}_k$.

The reason for introducing the compact form is to allow us to exploit some standard theory for fixed point iteration. In the one norm, the balancing problem can be viewed as an attempt to find the positive solution of the nonlinear equation

$$\mathcal{D}(\mathbf{x})\mathbf{A}\mathbf{x} = \mathbf{e}. \tag{14}$$

This can be rewritten in many ways. In particular, noting that $\mathcal{D}(\mathbf{x})\mathbf{e} = \mathcal{D}(\mathbf{A}\mathbf{x})^{-1}\mathbf{e}$, we can write

$$\mathbf{x} = \mathcal{D}(\mathbf{x})\mathbf{e} = \sqrt{\mathcal{D}(\mathbf{x})^2}\mathbf{e} = \sqrt{\mathcal{D}(\mathbf{x})\mathcal{D}(\mathbf{A}\mathbf{x})^{-1}}\mathbf{e} = \sqrt{\mathcal{D}(\mathbf{A}\mathbf{x})^{-1}\mathbf{x}}.$$

In other words (13) is a fixed point iteration for solving (14). Note that Sinkhorn–Knopp can also be framed in terms of a fixed point iteration for solving (14), although some care needs to be taken to ensure that the correct iterates are extracted [21].

The consequence of this discussion is that we can prove a result on the asymptotic convergence rate of the algorithm by bounding the norm of the Jacobian of $f(\mathbf{x}) = \sqrt{\mathcal{D}(\mathbf{A}\mathbf{x})^{-1}\mathbf{x}}$ in the environs of a fixed point. We restrict $\mathbf{x}$ so that it lies in the positive cone $\mathbb{R}^n_+$ to ensure that all our diagonal matrices are invertible, and we only take the square roots of positive numbers. Not only is $f(\mathbf{x})$ continuous but it is differentiable. Differentiating $f(\mathbf{x})$ term by term, we find that its Jacobian can be written as

$$J(\mathbf{x}) = -\frac{1}{2}\mathcal{D}(\mathbf{x})^{1/2}\mathcal{D}(\mathbf{A}\mathbf{x})^{-3/2}\mathbf{A} + \frac{1}{2}\mathcal{D}(\mathbf{A}\mathbf{x})^{-1/2}\mathcal{D}(\mathbf{x})^{-1/2}. \tag{15}$$

To establish the asymptotic rate of convergence, we want to bound $\|J(\mathbf{x})\|$ at the fixed point $\mathbf{x}_*$. Substituting the identity $\mathcal{D}(\mathbf{A}\mathbf{x}_*) = \mathcal{D}(\mathbf{x}_*)^{-1}$ into (15) gives

$$\begin{aligned}J(\mathbf{x}_*) &= -\frac{1}{2}\mathcal{D}(\mathbf{x}_*)^{1/2}\mathcal{D}(\mathbf{x}_*)^{3/2}\mathbf{A} + \frac{1}{2}\mathcal{D}(\mathbf{A}\mathbf{x}_*)^{1/2}\mathcal{D}(\mathbf{x}_*)^{-1/2} \\ &= \frac{1}{2}\left(\mathbf{I} - \mathcal{D}(\mathbf{x}_*)^2\mathbf{A}\right) = \frac{1}{2}\mathcal{D}(\mathbf{x}_*)\left(\mathbf{I} - \mathcal{D}(\mathbf{x}_*)\mathbf{A}\mathcal{D}(\mathbf{x}_*)\right)\mathcal{D}(\mathbf{x}_*)^{-1} \\ &= \frac{1}{2}\mathcal{D}(\mathbf{x}_*)(\mathbf{I} - \mathbf{P})\mathcal{D}(\mathbf{x}_*)^{-1},\end{aligned}$$

where $\mathbf{P}$ is symmetric and stochastic. In a neighbourhood of $\mathbf{x}_*$ we can bound the asymptotic rate of convergence of the iteration by the largest eigenvalue of $(\mathbf{I}-\mathbf{P})/2$. Since the spectrum of $\mathbf{P}$ lies in $[-1,1]$, the bound is $(1-\lambda_{\min}(\mathbf{P}))/2$.

We can couple our observations with the convergence results from Section 3 to make a strong statement about the performance of the algorithm.

Theorem 2. *Let $\mathbf{A} \geq 0$ be symmetric and be fully indecomposable. Then for any vector $\mathbf{x}_0 > 0$ the iteration (13) will converge to the unique positive vector $\mathbf{x}_*$ such that $\mathcal{D}(\mathbf{x}_*)\mathbf{A}\mathcal{D}(\mathbf{x}_*) = \mathbf{P}$ is doubly stochastic and the convergence is asymptotically linear at the rate $(1-\lambda_{\min}(\mathbf{P}))/2 < 1$.*

Proof. We already know that (13) converges from Theorem 1. The uniqueness of $\mathbf{x}_*$ is well known (see, for example, Lemma 4.1 in [21]).

Since $\mathbf{A}$ is fully indecomposable, so is $\mathbf{P}$. Such a doubly stochastic matrix is primitive [7] and hence it has a single simple eigenvalue of modulus 1. Therefore $\lambda_{\min}(\mathbf{P}) > -1$ and the asymptotic rate of convergence, established above, is bounded below one. □

Note that Theorem 1 establishes the convergence of (13) for matrices with total support. If a matrix has total support but is not fully indecomposable then it can be permuted (not necessarily symmetrically) into a block diagonal matrix where each block is fully indecomposable [25]. Suppose, then, that $\mathbf{A} \geq 0$ is symmetric and has total support and can be permuted into the form

$$\begin{pmatrix} \mathbf{A}_1 & & & \\ & \mathbf{A}_2 & & \\ & & \ddots & \\ & & & \mathbf{A}_k \end{pmatrix}$$

where each diagonal block is fully indecomposable. If the permutation is symmetric then we can apply Theorem 2 to each of the diagonal blocks in turn. If no such symmetric permutation exists then, because of the nonnegativity of $\mathbf{A}$, there must be a symmetric permutation into a block diagonal form where the blocks are either fully indecomposable or have the structure

$$\begin{pmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{pmatrix}$$

where $\mathbf{B}$ is fully indecomposable. We can still give the rate of convergence in this case: it is covered by a general convergence result for (13) on unsymmetric matrices, which we now establish.

3.4 Rate of convergence for unsymmetric matrices

Much of our analysis carries through if $\mathbf{A}$ is unsymmetric and we work with

$$\begin{pmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{pmatrix}$$

instead: using (12) on this matrix is exactly the same as running Algorithm 1 on $\mathbf{A}$. At a fixed point, the Jacobian matrix can be written

$$J(\mathbf{x}_*) = \frac{1}{2}\mathcal{D}(\mathbf{x}_*)\left(\mathbf{I} - \begin{pmatrix} \mathbf{0} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{pmatrix}\right)\mathcal{D}(\mathbf{x}_*)^{-1}, \tag{16}$$

where $\mathbf{P}$ is doubly stochastic but unsymmetric. There is a snag, though. Notice that

$$\begin{pmatrix} \mathbf{0} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{e} \\ -\mathbf{e} \end{pmatrix} = \begin{pmatrix} -\mathbf{e} \\ \mathbf{e} \end{pmatrix},$$

from which we conclude that -1 is an eigenvalue of $J(\mathbf{x}_*)$ and we can no longer apply the contraction mapping theorem. Nor is the fixed point unique: if $\mathbf{P} = \mathbf{DAE}$ then $\alpha\mathbf{D}$ and $\mathbf{E}/\alpha$ give another solution.

A similar problem arises in trying to apply the contraction mapping theorem in the analysis of the Sinkhorn–Knopp algorithm [21]. Here, as there, this difficulty is overcome once we note that we do not really care which fixed point we end up at, just that we get closer and closer to the set of fixed points. But we know this happens: we have a global convergence result (Theorem 1).

Theorem 3. *Let*

$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{pmatrix}$$

where $\mathbf{B} \in \mathbb{R}^{n\times n}$ *is nonnegative and fully indecomposable and let* $\mathbf{Q}$ *be the doubly stochastic matrix one gets from diagonally balancing* $\mathbf{B}$.

The iteration (13) converges linearly for all initial vectors $\mathbf{x}_0 > 0$ *with asymptotic rate of convergence* $\rho = (1 + \sigma_2(\mathbf{Q}))/2$ *where* $\sigma_2(\mathbf{Q})$ *is the second largest singular value of* $\mathbf{Q}$.

As with Theorem 2, we can extend Theorem 3 to cover unsymmetric matrices which are not fully indecomposable but have total support by working with a block diagonal permutation.

Proof. Convergence of the algorithm for any starting vector is guaranteed by Theorem 1. Let us assume that the limit point is $\mathbf{x}_* = [\mathbf{r}^T\ \mathbf{c}^T]^T$. Note that the set of fixed points of (13) can be written

$$S = \left\{ \begin{pmatrix} \beta\mathbf{r} \\ \frac{1}{\beta}\mathbf{c} \end{pmatrix},\ \beta \in \mathbb{R}_+ \right\}.$$

Let $\epsilon > 0$ and choose K so that for $k > K$, $\|\mathbf{x}_k - \mathbf{x}_*\| < \epsilon$ where we choose (for reasons we will explain)

$$\|\mathbf{x}\|_* = \sqrt{\frac{\mathbf{x}^T\mathcal{D}(\mathbf{x}_*)^{-2}\mathbf{x}}{2n}}.$$

We will show that

$$\min_{\mathbf{s}\in S} \|\mathbf{x}_{k+1} - \mathbf{s}\|_* \le \rho \min_{\mathbf{s}\in S} \|\mathbf{x}_k - \mathbf{s}\|_* + \tau_k, \tag{17}$$

where $\tau_k = \mathcal{O}(\epsilon^2)$, and hence we can infer the desired convergence rate for small enough values of ϵ. Note that $\mathbf{x}_*$ may not be the nearest element to $\mathbf{x}_k$ in S: the nearest point is

$$\mathbf{s}_k = \begin{pmatrix} \alpha\mathbf{r} \\ \frac{1}{\alpha}\mathbf{c} \end{pmatrix}$$

where $\alpha = 1 + \epsilon_\alpha$ and $\epsilon_\alpha = \mathcal{O}(\epsilon)$.

Now

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) = f(\mathbf{x}_* + \mathbf{p}) = \mathbf{x}_* + J(\mathbf{x}_*)\mathbf{p} + \mathbf{q},$$

where $\|\mathbf{p}\|_* < \epsilon$ and $\|\mathbf{q}\|_* = \mathcal{O}(\epsilon^2)$. From (16) we can deduce that the eigenvalues of $J(\mathbf{x}_*)$ are

$$\lambda_i = \frac{1}{2}(1 + \sigma_i(\mathbf{Q})), \quad \lambda_{i+n} = \frac{1}{2}(1 - \sigma_{n+1-i}(\mathbf{Q})), \quad i = 1, 2, \ldots, n,$$

where $\sigma_i(\mathbf{Q})$ is the ith singular value of $\mathbf{Q}$. Since $\mathbf{B}$ is fully indecomposable then so is $\mathbf{Q}$ and it is primitive. Hence

$$1 = \sigma_1(\mathbf{Q}) > \sigma_2(\mathbf{Q}) \geq \cdots \geq \sigma_n(\mathbf{Q}) \geq 0.$$

Let $\mathbf{v}_i$ be the eigenvector of $J(\mathbf{x}_*)$ associated with λ_i such that $\|\mathbf{v}_i\|_* = 1$ and write

$$\mathbf{p} = \sum_{i=1}^{2n} \mu_i \mathbf{v}_i.$$

From (16) we know that $J(\mathbf{x}_*)$ is similar to a symmetric matrix. Our choice of norm is motivated by our need to control the size of $J(\mathbf{x}_*)\mathbf{p}$ for all $k > K$. Since the $\mathbf{v}_i$ form an orthonormal set with respect to the inner product that induces our norm, $|\mu_i| \leq \|\mathbf{p}\|_* < \epsilon$, for all i.

Noting that $\lambda_1 = 1$, $\mathbf{v}_1 = [\mathbf{r}^T \ -\mathbf{c}^T]^T$, $\lambda_{2n} = 0$ and $\mathbf{v}_{2n} = \mathbf{x}_*$, we have

$$\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{x}_* + \mu_1 \mathbf{v}_1 + \sum_{i=2}^{2n-1} \lambda_i \mu_i \mathbf{v}_i + \mathbf{q} \\
&= \begin{pmatrix} (1+\mu_1)\mathbf{r} \\ (1-\mu_1)\mathbf{c} \end{pmatrix} + \sum_{i=2}^{2n-1} \lambda_i \mu_i \mathbf{v}_i + \mathbf{q} \\
&= \begin{pmatrix} \gamma\mathbf{r} \\ \mathbf{c}/\gamma \end{pmatrix} + \begin{pmatrix} 0 \\ \eta\mathbf{c} \end{pmatrix} + \sum_{i=2}^{2n-1} \lambda_i \mu_i \mathbf{v}_i + \mathbf{q}
\end{aligned}$$

where $\gamma = (1 + \mu_1) = 1 + \mathcal{O}(\epsilon)$ and $\eta = -\mu_1^2/(1 + \mu_1)$.

In order to establish (17) we need to relate the size of the sum in our expression for $\mathbf{x}_{k+1}$ to $\min_{\mathbf{s}\in S} \|\mathbf{x}_k - \mathbf{s}\|_* = \|\mathbf{x}_k - \mathbf{s}_k\|_*$. Since

$$\mathbf{x}_k - \mathbf{s}_k = \mathbf{p} + \mathbf{x}_* - \mathbf{s}_k = \mathbf{p} + \begin{pmatrix} (1-\alpha)\mathbf{r} \\ (1-1/\alpha)\mathbf{c} \end{pmatrix} = \sum_{i=1}^{2n} \mu_i \mathbf{v}_i + \frac{1-\alpha^2}{2\alpha}\mathbf{v}_1 - \frac{(\alpha-1)^2}{2\alpha}\mathbf{v}_{2n},$$

we have

$$\left\| \sum_{i=2}^{2n-1} \lambda_i \mu_i \mathbf{v}_i \right\|_* \leq \lambda_2 \sqrt{\sum_{i=2}^{2n-1} \|\mu_i \mathbf{v}_i\|_*^2} = \lambda_2 \left\| \sum_{i=2}^{2n-1} \mu_i \mathbf{v}_i \right\|_* \leq \lambda_2 \|\mathbf{x}_k - \mathbf{s}_k\|_*,$$

and letting

$$\tau_k = \left\| \begin{pmatrix} 0 \\ \eta\mathbf{c} \end{pmatrix} + \mathbf{q} \right\|_* = \mathcal{O}(\epsilon^2),$$

we are done. □

Our rate of convergence result allows a direct comparison with other scaling algorithms. In particular, in [21] it is shown that the Sinkhorn–Knopp algorithm also converges linearly with an asymptotic rate of convergence $\sigma_2(\mathbf{Q})$. Note that since $\sigma_2(\mathbf{Q}) < 1$, $\rho > \sigma_2(\mathbf{Q})$. Hence the bound on the asymptotic rate of convergence (and empirical evidence suggests that it is generally a sharp bound) is **necessarily** larger than the corresponding bound for the Sinkhorn–Knopp algorithm.

However, for symmetric matrices the comparison is not so clear cut. The asymptotic rates are $\lambda_2(\mathbf{P})$ for the Sinkhorn–Knopp algorithm versus $(1 - \lambda_{\min}(\mathbf{P}))/2$ for Algorithm 1. For sparse matrices we can expect both of these factors to be close to one (especially if the matrix is close to being reducible). But note that when $|\mathbf{A}|$ is symmetric positive definite, the rate of convergence of Algorithm 1 is guaranteed to be good, since $\mathbf{P}$ is symmetric positive definite, too.

Often, though, we only need a modest degree of scaling to induce beneficial effects in linear solvers. Here, the fact that Algorithm 1 retains symmetry is a distinct advantage: even though a symmetric matrix balanced by the Sinkhorn–Knopp algorithm will (in the limit) be symmetric, the intermediate scalings lose this property.

A number of balancing algorithms [22, 23] have been developed based on Newton iterations for solving (14). While these retain symmetry in the intermediate scalings and can achieve quadratic convergence asymptotically, they can behave erratically when the iterates are far from being truly balanced, a phenomenon we have not seen with Algorithm 1.

4 Numerical experiments

We have implemented the proposed algorithm in Matlab (version R2009a), and compared it against our implementation of Bunch's and Sinkhorn–Knopp algorithms. We have experimented with a set of matrices obtained from the University of Florida Sparse Matrix Collection (UFL) available at `http://www.cise.ufl.edu/research/sparse/matrices/`. The matrices in our data set satisfy the following properties: real, $1000 \leq n \leq 121000$, $2n \leq \text{nnz} \leq 1790000$, without explicit zeros, fully indecomposable, not a matrix with nonzeros only in $\{-1, 0, 1\}$. There were a total of 224 such matrices at the time of experimentation. In seven out of those 224 matrices, Matlab's `condest` function failed to compute an estimated condition number in a reasonable amount of time, and three matrices were singular (`condest` gave `inf` as the estimated condition number). Therefore, we give results on 214 matrices, among which 64 are unsymmetric, 46 are symmetric positive definite, and 104 are symmetric but not positive definite.

We present two different sets of experiments. In the first one, we compare different scaling algorithms. In the second one, we investigate the merits of using the proposed scaling algorithm in solving linear systems with a direct method. In terms of convergence, there is little difference in theory which p-norm we choose to scale by in Algorithm 1 (apart from the ∞-norm). But to investigate the potential of the algorithm in practice we have tested the both the 1-norm and 2-norm versions alongside the ∞-norm. For comparison, we have also tested the 1- and 2-norm versions of the Sinkhorn–Knopp algorithm as well as Bunch's

Table 1: The numbers of matrices in which the estimated condition number of the scaled matrix is larger than (the first half of the table) and smaller than (the second half) the condition number (estimated) of the original matrix by differing amounts, for the unsymmetric, symmetric, and symmetric positive definite matrices in the data set.

	Unsymmetric (64)				Symmetric (104)				SPD (46)			
ratio	B	A-inf	A-1	A-2	B	A-inf	A-1	A-2	B	A-inf	A-1	A-2
$\geq$ 1.0e+3	0	0	6	5	0	0	0	0	0	0	0	0
$\geq$ 1.0e+2	0	0	6	6	1	0	1	0	0	0	0	0
$\geq$ 1.0e+1	1	0	8	6	11	0	1	0	0	1	0	0
$>$ 1.0e+0	9	6	12	12	32	28	15	18	7	8	6	8
$\leq$ 1.0e-3	11	10	11	9	18	20	25	26	10	10	11	10
$\leq$ 1.0e-4	6	8	8	7	11	13	19	23	7	7	7	7
$\leq$ 1.0e-5	5	6	6	5	8	8	11	14	3	3	3	3
$\leq$ 1.0e-6	5	4	3	4	5	5	8	8	3	3	3	3

∞-norm algorithm.

4.1 Condition numbers and iterations

Here, we compare the proposed scaling algorithms with a few others.

First, we comment on the condition number estimates of the scaled matrices, where the proposed algorithm and Sinkhorn–Knopp algorithm are run with an error tolerance parameter of 1.0e-4. For a given scaling algorithm, we compute the scaled matrix and run Matlab's `condest` function on it. We used the performance profiles discussed in [15] to generate the plots in Fig. 1. Each plot is related to the condition estimates after the application of four different scaling algorithms and those of the original matrices for a given matrix type. For a given τ, the plot shows the fraction of the number of matrices for which a scaling algorithm obtains a condition estimate within e^τ, where e is the base of natural logarithm, of the best (among all five condition estimates). Therefore, the higher the fraction the more preferable the scaling method is. The plots for Sinkhorn–Knopp algorithm in 1- and 2-norms are not shown, as the estimated condition numbers agreed, on average, to the fourth significant digit with those of the proposed algorithm with the corresponding norms. As seen in the curves, the estimated condition number is almost always improved after the scaling algorithms. The ∞-norm scaling algorithms (both the proposed one and that of Bunch) are preferable to 1- and 2-norm scaling algorithms for unsymmetric matrices, and vice versa for general symmetric ones. For the symmetric positive definite matrices, all algorithms are almost equal if one accepts solutions that are within e of the best. We give further details in Table 1.

As seen from Table 1, all of the scaling algorithms improve the condition number in most of the instances. The highest increase for Bunch's algorithm in unsymmetric, symmetric, and symmetric positive definite matrices are, respectively, 1.36e+1, 1.16e+0, and 2.21e+0; for the proposed algorithm in the ∞-norm these numbers are 4.03e+0, 1.16e+0, and 1.36e+1. The proposed algorithm in 1 and 2-norms suffers in some of the unsymmetric matrices (problem ids at the UFL collection are 287, 289, 816, 1200, and 1433) where an increase in

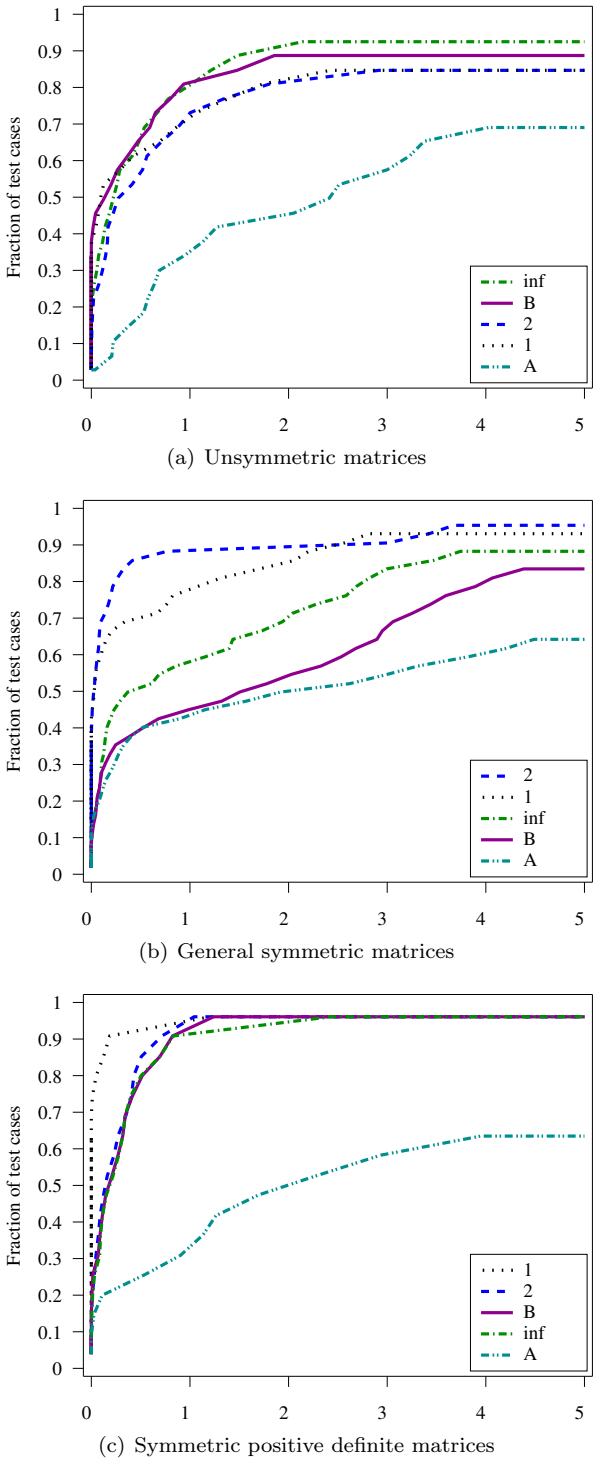


Figure 1: Performance profiles for the condition number estimates for unsymmetric matrices 1(a), general symmetric matrices 1(b), and symmetric positive definite matrices 1(c). `A` marks the condition number estimate of the original matrix; `B` marks that of Bunch's algorithm; `inf`, `1`, and `2` mark that of the proposed algorithm with ∞-, 1-, and 2-norms.

Table 2: The statistics on the numbers of iterations of Sinkhorn–Knopp algorithm in 1- and 2-norms (SK-1 and SK-2) and the proposed algorithm in 1-, 2-, and ∞-norms (A-1, A-2, and A-∞) to reach an error tolerance $\varepsilon = 1.0e\text{-}4$.

matrix type	statistics	SK-1	SK-2	A-1	A-2	A-∞
unsymmetric	min	1	47	1	6	2
	avg	9715	17066	23838	38303	9
	med	2135	4905	2436	4897	8
	max	116205	177053	307672	519249	19
symmetric	min	8	1	3	1	2
	avg	880	2940	431	1110	10
	med	238	700	32	33	13
	max	11870	22302	10307	18925	19
sym pos def	min	73	46	7	3	2
	avg	1614	3270	13	11	6
	med	444	1494	14	12	3
	max	11271	14418	17	18	17

the order of the 10^{34} was observed with the 1-norm algorithm for the matrix 287 (Sinkhorn–Knopp algorithm had an increase in the order of 10^{36} for the same matrix) and an increase in the order of 1.0e+12 was observed with the 2-norm algorithm for the same matrix (Sinkhorn–Knopp algorithm in 2-norm obtained an increase in the order of 1.0e+14). On the other hand, the proposed algorithm in 1- and 2-norms always reduced the condition number for general symmetric matrices (the highest ratios of the condition numbers to those of the matrices are 5.97e-1 and 8.73e-1, respectively, for the two norms). For the symmetric positive definite matrices, the highest numbers are 5.67e+0 and 1.28e+0 for the proposed algorithm in 1- and 2-norms. As seen from these numbers, using the proposed algorithm in 1- and 2-norms for unsymmetric matrices can worsen the condition numbers dearly, whereas it seems always safe to use them in other types of matrices, as also foretold by our theoretical investigations of Section 3.

We present some statistical indicators (the minimum, the average, the median, and the maximum) of the number of iterations for convergence with a tolerance of 1.0e-4, that is ε =1.0e-4 in (3), for Sinkhorn–Knopp algorithm in 1- and 2-norms, and for the proposed algorithm in 1-, 2-, and ∞-norms in Table 2. First we comment on the proposed algorithm in the ∞-norm. As seen from the numbers, the algorithm converges in a dozen of iterations. We note that the interesting case is the symmetric matrices (for the unsymmetric ones, an algorithm of type Sinkhorn–Knopp converges in one iteration, and for the symmetric positive definite ones, scaling with the reciprocal of the square roots of the diagonal elements on both sides will have the desired effect). The median of the number of iterations is between the average and the maximum, displaying a typical number of iterations around 13.

As seen in Table 2, the 1-norm and 2-norm scaling algorithms have large number of iterations for unsymmetric matrices. On average, Sinkhorn–Knopp algorithm looks much better than the corresponding proposed algorithm with the median values being close to each other. This shows that the proposed algorithm has some difficulties with a few instances (this is also confirmed by

the huge difference in the maximum number of iterations and the median values). For the symmetric matrices, the proposed algorithm is much better than Sinkhorn–Knopp algorithm in any of the presented statistical indicators. The median number of iterations for the proposed algorithm in 1-norm is 32 which is much better than that of SK-1. Again, there are a few outliers, as the average is much higher than the median and the maximum is in turn much higher than the average. The numbers of iterations (for symmetric matrices) of A-1 and A-2 that are larger than the average values are, respectively, 15 and 18 (out of 104), also confirming the existence of only a few outliers. For the symmetric positive definite matrices, the proposed algorithm in 1- and 2-norms converge very fast compared to SK-1 and SK-2. These latter two algorithms have a comportment on symmetric positive definite matrices similar to what they have for general symmetric matrices and have larger number of iterations.

4.2 Uses of the methods in a direct solver

We now comment on the use of the proposed scaling algorithm in the solution of linear systems with direct methods. We use the solver MUMPS [3, 4] version 4.10 to test the effects of the proposed scaling method. We use MUMPS with default parameter settings except for the following. For the unsymmetric matrices, if there are zeros in the diagonal, we find a maximum traversal using `dmperm` of Matlab and permute the matrix to put that transversal into the diagonal before calling MUMPS and set the related parameter ICNTL(6) to 0 in MUMPS. For the general symmetric matrices, we set the parameter ICNTL(12) to 2 to use the compressed ordering strategies discussed in [17] along with a columnwise permutation option ICNTL(6)=1, equivalent in essence to `dmperm` of Matlab. We set ICNTL(14) to 80 (a memory relaxation parameter that permits the use of 80% more memory than estimated), MUMPS returns an error message for factorizations that passes this limit. We further deemed any factorization in which MUMPS returned a warning as unsuccessful (this happened only for the matrix with id 2569). The necessity of more memory than the estimated amount arises when pivoting is performed for the numerical stability. A scaling strategy deemed to be better, if it increases the number of successful runs by reducing pivoting. We do not perform experiments with symmetric positive matrices.

As the number of iterations of the proposed algorithm can be high (see Table 2, where even an average 32 iterations for symmetric matrices can be considered as high in the context of direct solvers), we do not recommend the use of the proposed algorithm till convergence. We rather recommend its use with a fixed and moderate number of iterations. In order to suggest a solid scaling approach, we have investigated different strategies. These scaling strategies can be specified by three integers $[i_1, i_2, i_3]$. Each scaling strategy will proceed in three phases: in phase 1, it will apply i_1 many steps of ∞-norm scaling; then in phase 2, it will apply i_2 many steps of 1- or 2-norm scaling; and finally in phase 3, it will apply i_3 many steps of ∞-norm scaling. It is understood that each phase continues using the scaling factors found so far, and if convergence occurs within a phase then the next phase is started. We use the ∞-norm scaling in the first phase as a smoother which reduces each nonzero entry below one. The 1- and 2-norm scaling iterations are two different alternatives leading to two different scaled matrices; we do not therefore use one after another. The ∞-norm scaling at phase 3 is also used as a smoother, which sets the maximum element in a row

Table 3: The effects of different scaling strategies on the direct solver MUMPS version 4.10. A strategy is said to be unsuccessful (column usuc.), if MUMPS returns a warning or an error message with the settings described in the text. The geometric mean of the ratio of the actual memory used (by MUMPS) to the estimated memory is given in the column $\frac{actM}{estM}$; the geometric mean of the ratio of the condition number estimate of the scaled matrix to the original one is given in the column $\frac{cnd(DAE)}{cnd(A)}$; the median of the number of off-diagonal pivots (whenever it is nonzero) is given in the column off-piv.

	unsymmetric matrices				general symmetric matrices			
strategy	usuc.	$\frac{actM}{estM}$	$\frac{cnd(DAE)}{cnd(A)}$	off-piv.	usuc.	$\frac{actM}{estM}$	$\frac{cnd(DAE)}{cnd(A)}$	off-piv.
no-scaling	7	1.02	6.20e-01	308	23	1.09	2.70e+01	3454
$[0, 3^{(1)}, 0]$	3	1.02	1.23e-03	67	4	1.04	1.89e-03	3458
$[1, 3^{(1)}, 0]$	3	1.01	1.17e-03	44	3	1.04	1.83e-03	3454
$[1, 3^{(1)}, 1]$	3	1.01	3.85e-04	45	3	1.03	1.08e-04	3454
$[1, 3^{(1)}, 3]$	3	1.01	3.80e-04	53	3	1.03	1.16e-04	3454
$[0, 10^{(1)}, 0]$	3	1.01	1.54e-03	157	1	1.03	3.94e-03	3462
$[1, 10^{(1)}, 0]$	3	1.01	1.54e-03	160	1	1.03	3.86e-03	3462
$[1, 10^{(1)}, 1]$	2	1.02	8.09e-04	66	1	1.03	1.70e-04	3462
$[1, 100^{(1)}, 0]$	3	1.01	1.21e-05	148	0	1.05	4.97e-03	3580
$[0, 3^{(2)}, 0]$	0	1.02	3.78e-02	8	3	1.01	2.26e-02	5504
$[1, 3^{(2)}, 0]$	0	1.01	3.54e-02	9	2	1.02	1.74e-02	5504
$[1, 3^{(2)}, 1]$	0	1.01	3.88e-02	9	2	1.02	2.03e-02	5504
$[1, 3^{(2)}, 3]$	0	1.01	4.05e-02	9	2	1.02	2.17e-02	5504
$[0, 10^{(2)}, 0]$	0	1.01	3.21e-02	8	1	1.02	1.32e-02	5504
$[1, 10^{(2)}, 0]$	0	1.01	3.39e-02	8	1	1.02	1.36e-02	5504
$[1, 10^{(2)}, 1]$	0	1.01	3.56e-02	8	1	1.02	1.45e-02	5504
$[1, 100^{(2)}, 0]$	0	1.01	3.50e-02	8	2	1.01	1.71e-02	5504
Bunch					1	1.03	5.83e-02	5504
SK10	0	1.01	3.52e-02	9				

and column close to one. The results are presented in Table 3. In this table, the scaling strategies are represented as $[i_1, i_2^{(1)}, i_3]$ or $[i_1, i_2^{(2)}, i_3]$ where $i_2^{(1)}$ and $i_2^{(2)}$ designate the number of 1-norm or 2-norm iterations performed in phase 2 of the scaling strategy. We run Bunch's algorithm only for general symmetric matrices, and 10 iterations of Sinkhorn–Knopp algorithm (shown with SK10) only for unsymmetric matrices. We note that MUMPS can run to completion for all matrices in our data set with with one of the scaling strategies. Therefore any error message returned by MUMPS signifies difficulties.

We comment more on symmetric matrices, as the proposed algorithm is particularly motivated for this case. As seen in the right half of Table 3, the proposed algorithm reduces the number of unsuccessful runs from 23 to 4 with only a few iterations. In general, the higher the number of iterations, the better the success rate is. The strategies with the 2-norm iterations look more effective in this sense with small number of iterations. The number of off-diagonal pivots are reduced more by the strategies having 1-norm iterations than those having the 2-norm iterations (we have also looked at the averages of all values, rather

than median of a few nonzero values of the number of off-diagonal pivots and observed the same outcome). The condition number estimates are better with the scaling strategies having 1-norm iterations; this is in correlation with the number of off-diagonal pivots (we note that the estimates in Matlab are done for 1-norm, so this may also have some effect). In the cases where the direct solver had success in all symmetric matrices, the condition numbers and the number of pivots are better with 1-norm iterations than the two-norm iterations. The geometric mean of the actual memory requirement over the estimated one is very modest in all cases (note that this value is computed for successful runs only). The scaling strategy $[0, 3^{(1)}, 0]$ or $[1, 3^{(1)}, 0]$ or their equivalent with 2-norm are the best alternatives for the direct solvers (to reduce the number of unsuccessful runs and to reduce the need for off-diagonal pivoting without too much cost).

For the unsymmetric matrices, the strategies with the 2-norm iterations are better than the 1-norm iterations in reducing the unsuccessful runs and the off-diagonal pivots, while being inferior for the condition estimates. Again, the memoir increase is with respect to the estimated amount is modest. The performance of SK10 for unsymmetric matrices and Bunch's algorithm for symmetric matrices in terms of the number of successful runs is remarkable(also remarkable is the number of off-diagonal pivots). Since Bunch's algorithm is sequential in nature, we conclude that for symmetric matrices on a sequential execution environment, it is the best alternative. For unsymmetric matrices, Sinkhorn–Knopp's algorithm is preferable to the proposed one. A few iterations of the proposed algorithm with a strategy $[0, 3^{(1)}, 0]$ or $[1, 3^{(1)}, 0]$ are preferable for symmetric matrices in parallel computing environments, as this strategy reduces the need for pivoting and the number of unsuccessful runs, while being parallelizable.

5 Conclusion

We presented an iterative algorithm which scales the ∞-norm of the rows and columns of a matrix to 1. The important features of the proposed algorithm are the following: it preserves symmetry; it is permutation independent; and it has fast linear convergence with an asymptotic rate of $1/2$. We discussed the extension to the 1-norm in detail. Again, the algorithm preserves symmetry and (trivially) is permutation independent. From the various theorems collecting the convergence analysis results in the different norms, it can be seen that the assumptions are much more restrictive and complicated in their combinatorial aspects in the 1-norm as opposed to the ∞-norm. In the 1-norm, we have established that convergence is dependent on the nonzero structure of the matrix in much the same way as other Diagonal Product Increasing algorithms, such as Sinkhorn–Knopp; but that for symmetric matrices our algorithm can accelerate this convergence. This is confirmed in experimental results.

The rates of convergence in the particular case of our algorithm, fixed and fast in the ∞-norm, problem-dependent and potentially much slower in the 1-norm, also illustrate the strong difference between the two. But whatever norm we choose, we have numerical evidence for the algorithm's potential for improving the condition number of a matrix by scaling. And our algorithm appears more reliable than existing alternatives, particularly for symmetric matrices.

Furthermore, combining two types of scaling can offer advantages over a single method. We have demonstrated numerical experiments in which the proposed algorithm was shown to be helpful for a direct solver in the sense that it reduces the need for numerical pivoting. In particular we have shown that one step of ∞-norm scaling followed by a few steps of 1- or 2-norm scaling is usually good enough. We have also experimentally demonstrated that the proposed algorithm usually converges faster than a known alternative for symmetric matrices.

Acknowledgements and availability

We thank Patrick R. Amestoy, Iain S. Duff, Nick Gould, and Mario Arioli for commenting on an earlier draft of the article. We also wish to thank the referees for their detailed reports that helped us improve the manuscript, and in particular Stanley Eisenstat for his very constructive editorial and technical comments.

The Matlab implementations of all scaling algorithms discussed in this paper, as well as an MPI-based parallel implementation of the proposed scaling algorithm, are available at `http://perso.ens-lyon.fr/~bucar/software.html`. The MPI-based parallel version is incorporated in MUMPS [3, 4] (since version 4.8) for scaling distributed input matrices. The sequential fortran implementation of the proposed scaling algorithm is available as `MC77` in the HSL Mathematical Software Library (`http://www.hsl.rl.ac.uk/`).

A General convergence results in the 1-norm

Parlett and Landis [24] give a general convergence result for scaling algorithms based on three specific properties. These properties ensure convergence when a given scaling algorithm produces a sequence of iterates that fulfill them. Such algorithms are called "*Diagonal Product Increasing algorithms*" (DPI). In this appendix, we extend partly the general convergence theorem of Parlett and Landis in that we show that some of their hypothesis can be weakened while keeping the same results, and that under some stronger hypothesis, their general convergence result can be strengthened too.

For the notations, consider a given iterative scaling algorithm that produces a sequence of matrices diagonally equivalent to the starting matrix $\mathbf{A} = \mathbf{A}^{(0)}$ with the following iterates:

$$\begin{aligned} \mathbf{A}^{(k)} &= \left(a_{ij}^{(k)}\right) = \mathbf{D}^{(k)}\mathbf{A}\mathbf{E}^{(k)}, \quad k = 1, 2, \ldots, \\ \mathbf{D}^{(k)} &= \operatorname{diag}\left(d_1^{(k)}, \ldots, d_n^{(k)}\right), \\ \mathbf{E}^{(k)} &= \operatorname{diag}\left(e_1^{(k)}, \ldots, e_n^{(k)}\right), \end{aligned} \tag{18}$$

where $\mathbf{D}^{(0)} = \mathbf{E}^{(0)} = \mathbf{I}$. For notational convenience let $r_i^{(k)}$, $i = 1, \ldots, n$, and

$c_j^{(k)}$, $j = 1, \ldots, n$, denote the 1-norm of rows and columns respectively, thus:

$$r_i^{(k)} = \|\mathbf{r}_i^{(k)}\|_1 = \sum_{j=1}^{n} |a_{ij}^{(k)}|,$$
$$c_j^{(k)} = \|\mathbf{c}_j^{(k)}\|_1 = \sum_{i=1}^{n} |a_{ij}^{(k)}|.$$

We also assume for simplicity that $\mathbf{A} \geq 0$, since scaling $\mathbf{A}$ or $|\mathbf{A}|$ will be the same. Under this simplification, the 1-norm of the rows and columns reduces to the row and column sums respectively, $r_i^{(k)} = \sum_{j=1}^{n} a_{ij}^{(k)}$ and $c_j^{(k)} = \sum_{i=1}^{n} a_{ij}^{(k)}$, and to generalize our results to any matrix, one just needs to extend the definition of a doubly stochastic matrix so that the absolute value of the matrix under consideration is doubly stochastic in the usual sense.

Theorem 4. *Suppose that a given scaling algorithm produces a sequence (18) of diagonal equivalents for a nonnegative matrix* $\mathbf{A}$ *that satisfy the three following properties:*

(P1) the sequence $(s_k)_{k\geq 1}$ *of the product of both the row and column scaling factors*

$$s_k = \prod_{i=1}^{n} d_i^{(k)} e_i^{(k)}, \quad k = 1, 2, \ldots$$

is monotonically increasing,

(P2) the 1-norm of rows and columns respectively converge to 1

$$\lim_{k\to\infty} r_i^{(k)} = 1 \text{ and } \lim_{k\to\infty} c_j^{(k)} = 1, \quad 1 \leq i, j \leq n \ ,$$

(P3) the sum of all the elements in the sequence of scaled equivalents for $\mathbf{A}$ *is bounded above*

$$\exists \beta > 0 \ s.t. \ \forall k \geq 1, \ \sum_{i=1}^{n} r_i^{(k+1)} = \sum_{j=1}^{n} c_j^{(k+1)} = \sum_{1\leq i,j\leq n} a_{ij}^{(k+1)} \leq \beta \ .$$

Then, if $\mathbf{A}$ *has support,* $\widehat{\mathbf{A}} = \lim_{k\to\infty} \mathbf{A}^{(k)}$ *exists and is doubly stochastic. Additionally, if* $\mathbf{A}$ *has total support, then this limit is diagonally equivalent to* $\mathbf{A}$*.*

Before getting into the details of the demonstration, which follows closely the developments made by Parlett and Landis for their main convergence theorem ([24, Theorem 1, pages 63–68]), we would like to mention that property (P1) is exactly the same as in the hypothesis they make, and that only (P2) and (P3) differ slightly in that they are partially weakened but sufficient (the three combined together) to raise the above conclusions, which are exactly the same as in [24, Theorem 1].

For the demonstration itself, we first establish the following intermediate result.

Lemma 1. *Using the same notations as in Theorem 4 above, property (P1) and property (P3) together (without property (P2)), as well as the fact that* $\mathbf{A}$ *has support, imply that*

1. $\lim_{k\to\infty} s_k = \xi > 0$ *exists,*
2. *there exists a strictly positive constant* γ *such that*

$$d_i^{(k)} e_j^{(k)} \geq \gamma$$

for all $k \geq 1$ *and for each index pair* (i,j) *such that* a_{ij} *is strictly positive and can be permuted onto a positive diagonal.*

Indeed, since $\mathbf{A}$ has support, there exists a permutation σ such that $a_{i,\sigma(i)} > 0$, $1 \leq i \leq n$. Let $a = \min_{1\leq i\leq n} (a_{i,\sigma(i)})$. Then, for all $k \geq 1$,

$$\sum_{i=1}^{n} d_i^{(k)} e_{\sigma(i)}^{(k)} a \leq \sum_{i=1}^{n} d_i^{(k)} e_{\sigma(i)}^{(k)} a_{i,\sigma(i)} = \sum_{i=1}^{n} a_{i,\sigma(i)}^{(k)} \leq \beta \,,$$

the last inequality resulting from (P3). Then, by the arithmetic-geometric mean inequality (8),

$$\forall k \geq 1, \;\; s_k = \prod_{i=1}^{n} \left(d_i^{(k)} e_{\sigma(i)}^{(k)} \right) \leq \left(\frac{1}{n} \sum_{i=1}^{n} d_i^{(k)} e_{\sigma(i)}^{(k)} \right)^n \leq \left(\frac{\beta}{n\, a} \right)^n \,,$$

and, combined with (P1), the monotonically increasing sequence $(s_k)_{k\geq 1}$ is thus bounded. Consequently,

$$\lim_{k\to\infty} s_k = \xi > 0$$

exists and is finite.

The demonstration of the second conclusion in Lemma 1 goes around the same ideas. For any $a_{ij} > 0$ that can be permuted onto a positive diagonal, there exists a permutation σ such that $j = \sigma(i)$ and $a_{l,\sigma(l)} > 0$, $1 \leq l \leq n$. Then, by (P1):

$$\forall k \geq 1, \;\; d_i^{(k)} e_j^{(k)} \prod_{\ell=1//\ell\neq i}^{n} \left(d_\ell^{(k)} e_{\sigma(\ell)}^{(k)} \right) = s_k \geq s_1 \,,$$

and

$$\forall k \geq 1, \;\; d_i^{(k)} e_j^{(k)} \geq s_1 \prod_{\ell=1//\ell\neq i}^{n} \left(d_\ell^{(k)} e_{\sigma(\ell)}^{(k)} \right)^{-1} \,.$$

Let $a = \min_{1\leq i,j\leq n} (a_{ij} > 0)$. Then, for all $k \geq 1$,

$$\sum_{\ell=1//\ell\neq i}^{n} d_\ell^{(k)} e_{\sigma(\ell)}^{(k)} a \leq \sum_{\ell=1//\ell\neq i}^{n} d_\ell^{(k)} e_{\sigma(\ell)}^{(k)} a_{\ell,\sigma(\ell)} = \sum_{\ell=1//\ell\neq i}^{n} a_{\ell,\sigma(\ell)}^{(k)} \leq \beta \,,$$

and, by the arithmetic-geometric mean inequality (8), we can conclude that

$$\forall k \geq 1, \;\; d_i^{(k)} e_j^{(k)} \geq s_1 \left(\frac{(n-1)a}{\beta} \right)^{n-1} = \gamma > 0 \,.$$

Proof of Theorem 4: Since, from (P3), all $a_{ij}^{(k)}$ are less than β after the first iteration, the sequence of matrices $(\mathbf{A}^{(k)})_{k\geq 1}$ is bounded in the finite dimensional space of real, $n \times n$ matrices. Consider any convergent subsequence $(\mathbf{A}^{(q)})_q$ of $(\mathbf{A}^{(k)})_{k\geq 1}$, and define

$$\widehat{\mathbf{A}} = (\hat{a}_{ij}) = \lim_{q\to\infty} \mathbf{A}^{(q)} \ .$$

From (P2),

$$\lim_{q\to\infty} r_i^{(q)} = 1 \ \text{ and } \ \lim_{q\to\infty} c_j^{(q)} = 1, \ \ 1 \leq i,j \leq n \ ,$$

which implies that $\widehat{\mathbf{A}}$ is doubly stochastic. Then, since the set of $n \times n$ doubly stochastic matrices is the convex hull of the set of $n \times n$ permutation matrices (see [8]), $\widehat{\mathbf{A}}$ must thus have total support. Therefore, on the one side, $\hat{a}_{ij} = \lim_{q\to\infty} a_{ij}^{(q)} = 0$ whenever $a_{ij} > 0$ cannot be permuted onto a positive diagonal. On the other side, consider any entry $a_{ij} > 0$ in $\mathbf{A}$ that can be permuted onto a positive diagonal, and let

$$\mu_{ij} = \frac{\hat{a}_{ij}}{a_{ij}} = \lim_{q\to\infty} d_i^{(q)} e_j^{(q)} \ .$$

From point 2 in Lemma 1, we know that $\mu_{ij} \geq \gamma > 0$.

Then, applying Lemma 2 of [24] (which is itself paraphrased from [31, page 345]), we know that there exist positive sequences $(x_i^{(q)})_q$ and $(y_i^{(q)})_q$ both with positive limits such that

$$d_i^{(q)} e_j^{(q)} = x_i^{(q)} y_j^{(q)}, \quad \forall \hat{a}_{ij} > 0 \text{ in } \widehat{\mathbf{A}}, \ \text{ and } \forall q \geq 1 \ .$$

Then, taking

$$\begin{aligned}
\mathbf{X}^{(q)} &= \operatorname{diag}\left(x_1^{(q)}, \ldots, x_n^{(q)}\right) \ , \\
\mathbf{Y}^{(q)} &= \operatorname{diag}\left(y_1^{(q)}, \ldots, y_n^{(q)}\right) \ , \\
\mathbf{X} &= \lim_{q\to\infty} \mathbf{X}^{(q)} \ \text{ and } \ \mathbf{Y} = \lim_{q\to\infty} \mathbf{Y}^{(q)} \ ,
\end{aligned}$$

we have

$$\widehat{\mathbf{A}} = \lim_{q\to\infty} \mathbf{D}^{(q)} \mathbf{S} \mathbf{E}^{(q)} = \lim_{q\to\infty} \mathbf{X}^{(q)} \mathbf{S} \mathbf{Y}^{(q)} = \mathbf{X}\mathbf{S}\mathbf{Y} \ ,$$

showing that $\widehat{\mathbf{A}}$ is diagonally equivalent to $\mathbf{S}$, where $\mathbf{S}$ is the submatrix with total support extracted from $\mathbf{A}$ where all entries $a_{ij} > 0$ in $\mathbf{A}$ that cannot be permuted onto a positive diagonal have been set to zero, the others remaining the same. Finally, if we consider any other convergent subsequence in $(\mathbf{A}^{(k)})_{k\geq 1}$, for the same reasons as above its limit will also be doubly stochastic and diagonally equivalent to $\mathbf{S}$, and since doubly stochastic equivalents are unique (see [32]), the two limits must be the same. Therefore, we can conclude that $\lim_{k\to\infty} \mathbf{A}^{(k)}$ exists and is doubly stochastic, and if additionally $\mathbf{A}$ has total support, then $\mathbf{A} = \mathbf{S}$ and is directly diagonally equivalent to the doubly stochastic matrix $\widehat{\mathbf{A}}$, which completes the proof of Theorem 4. ∎

The first thing to mention is that the previous three properties (P1), (P2), and (P3) are insufficient to prove that the two sequences of scaling matrices $\mathbf{D}^{(k)}$ and $\mathbf{E}^{(k)}$ actually converge. The reason why this additional result holds for Algorithm 1 is that, not only is the sequence $(s_k)_{k\geq 1}$ monotonically increasing,

but also both sequences $(\prod_{i=1}^{n} d_i^{(k)})_{k\geq 1}$ and $(\prod_{i=1}^{n} e_i^{(k)})_{k\geq 1}$ are independently monotonically increasing. We shall now extend the conclusions in Theorem 4 while strengthening some of the hypothesis that are made in the three generic properties (P1), (P2) and (P3) of being "*Diagonal Product Increasing*". The following theorem and corollary summarize this extension, and actually cover directly the convergence results that we have raised for our Algorithm 1 in section 3.

Theorem 5. *Suppose that a given scaling algorithm produces a sequence (18) of diagonal equivalents for a nonnegative matrix* $\mathbf{A}$*, each satisfying the following three properties:*

(P1) both sequences of rows and column scaling factors products*

$$\prod_{i=1}^{n} d_i^{(k)} \quad \textit{and} \quad \prod_{i=1}^{n} e_i^{(k)}, \ k = 1, 2, \ldots$$

are monotonically increasing (which is more strict and implies property (P1) above)

(P2) and (P3), which are the same as in Theorem 4 above.

Then, if $\mathbf{A}$ *is fully indecomposable, both limits* $\mathbf{D} = \lim_{k\to\infty} \mathbf{D}^{(k)}$ *and* $\mathbf{E} = \lim_{k\to\infty} \mathbf{E}^{(k)}$ *do exist and* $\widehat{\mathbf{A}} = \mathbf{DAE}$ *is doubly stochastic.*

Proof of Theorem 5: Since (P1*) implies (P1), and since (P2) and (P3) remain the same, we already know from the conclusions in Theorem 4 that if $\mathbf{A}$ has total support, $\widehat{\mathbf{A}} = \lim_{k\to\infty} \mathbf{A}^{(k)}$ exists and is diagonally equivalent to $\mathbf{A}$. To prove that both $\mathbf{D} = \lim_{k\to\infty} \mathbf{D}^{(k)}$ and $\mathbf{E} = \lim_{k\to\infty} \mathbf{E}^{(k)}$ exist and $\widehat{\mathbf{A}} = \mathbf{DAE}$, we shall exploit the stronger assumption that $\mathbf{A}$ not only has total support but is fully indecomposable. In this case, $\mathbf{A}$ is not only diagonally equivalent to the doubly stochastic limit $\widehat{\mathbf{A}} = \lim_{k\to\infty} \mathbf{A}^{(k)}$, but we also know from [31] that the diagonal matrices which take place in this equivalence are unique up to a scalar factor.

Let us suppose now that one of the sequences $(d_i^{(k)})_{k\geq 1}$ is unbounded for some i. In such a case, there exist a subsequence $(d_i^{(q)})_q$ such that

$$\lim_{q\to\infty} d_i^{(q)} = +\infty \ .$$

As the matrix $\mathbf{A}$ is fully indecomposable, for any index j, $1 \leq j \leq n$, there exist a chain of positive entries in which the row and column indexes alternately change

$$a_{ij_1}, a_{i_1 j_1}, a_{i_1 j_2}, a_{i_2 j_2}, a_{i_2 j_3}, \ldots, a_{i_{p-1} j_{p-1}}, a_{i_{p-1} j_p}, a_{i_p j_p}, a_{i_p j}$$

"*connecting*" row i to column j (see, for example, [11, Theorem 4.2.7]). Consequently, because of the conclusions raised at point 2 in Lemma 1, we know that the denominator in the following fraction

$$\frac{d_i^{(k)} e_{j_1}^{(k)} \, d_{i_1}^{(k)} e_{j_2}^{(k)} \, d_{i_2}^{(k)} e_{j_3}^{(k)} \ \ldots \ d_{i_{p-2}}^{(k)} e_{j_{p-1}}^{(k)} \, d_{i_{p-1}}^{(k)} e_{j_p}^{(k)} \, d_{i_p}^{(k)} e_j^{(k)}}{d_{i_1}^{(k)} e_{j_1}^{(k)} \, d_{i_2}^{(k)} e_{j_2}^{(k)} \ \ldots \ d_{i_{p-1}}^{(k)} e_{j_{p-1}}^{(k)} \, d_{i_p}^{(k)} e_{j_p}^{(k)}} \ = \ d_i^{(k)} e_j^{(k)}$$

is bounded away from zero and, from the demonstration of Theorem 4 above, that this fraction has a strictly positive limit, i.e. :

$$\lim_{k\to\infty} d_i^{(k)} e_j^{(k)} = \frac{\mu_{ij_1}\,\mu_{i_1j_2}\,\mu_{i_2j_3}\,\cdots\,\mu_{i_{p-2}j_{p-1}}\,\mu_{i_{p-1}j_p}\,\mu_{i_pj}}{\mu_{i_1j_1}\,\mu_{i_2j_2}\,\cdots\,\mu_{i_{p-1}j_{p-1}}\,\mu_{i_pj_p}} = \mu_{ij} > 0\,.$$

Since this can be done for any j, we can conclude that the subsequence $(\prod_{j=1}^n e_j^{(q)})_q$ (which is strictly positive) goes to zero as $(d_i^{(q)})^{-n}\prod_{j=1}^n \mu_{ij}$. This last conclusion is in contradiction with property (P1*) above, stating that both sequences $(\prod_{j=1}^n d_j^{(k)})_{k\geq 1}$ and $(\prod_{j=1}^n e_j^{(k)})_{k\geq 1}$ are monotonically increasing. The same could be done with one of the $(e_j^{(k)})_{k\geq 1}$ instead, and we can finally conclude that each of the sequences $(d_i^{(k)})_{k\geq 1}$ and $(e_i^{(k)})_{k\geq 1}$ are bounded.

Now, since the two sequences $(\mathbf{D}^{(k)})_{k\geq 1}$ and $(\mathbf{E}^{(k)})_{k\geq 1}$ are bounded in the finite dimensional space of real $n\times n$ diagonal matrices, they have convergent subsequences. Let us consider two convergent subsequences:

$$\left(\mathbf{D}^{(p)},\mathbf{E}^{(p)}\right) \underset{p\to+\infty}{\longrightarrow} (\mathbf{D}_1,\mathbf{E}_1)\,,$$

and

$$\left(\mathbf{D}^{(q)},\mathbf{E}^{(q)}\right) \underset{q\to+\infty}{\longrightarrow} (\mathbf{D}_2,\mathbf{E}_2)\,.$$

Obviously, $\mathbf{D}_1\mathbf{A}\mathbf{E}_1 = \widehat{\mathbf{A}} = \mathbf{D}_2\mathbf{A}\mathbf{E}_2$, and thus, because of the uniqueness (shown in [31]), there exists $\alpha > 0$ such that $\mathbf{D}_1 = \alpha\mathbf{D}_2$ and $\mathbf{E}_1 = (1/\alpha)\mathbf{E}_2$. Then, as mentioned above, since both sequences $\left(\prod_{i=1}^n d_i^{(k)}\right)_{k\geq 1}$ and $\left(\prod_{i=1}^n e_i^{(k)}\right)_{k\geq 1}$ are monotonically increasing, it is clear that α must be equal to 1 and the two limits must be equal. Therefore, we can conclude that $\mathbf{D} = \lim_{k\to\infty}\mathbf{D}^{(k)}$ and $\mathbf{E} = \lim_{k\to\infty}\mathbf{E}^{(k)}$ exist, which completes the proof of Theorem 5. ∎

Finally, we know from [25] that matrices with total support can be permuted into a direct sum of fully indecomposable matrices. Now, if the algorithm generates iterates (18) that are insensitive to permutations on the matrix $\mathbf{A}$ and, if on top of that applying it to a block diagonal matrix is equivalent to working directly which each diagonal block separately, we can easily extend the conclusions of Theorem 5 to the case of matrices with total support by collecting the conclusions from Theorem 5 independently on each fully indecomposable sub-matrix in such a direct sum. This is summarized in the following corollary.

Corollary 2. *Suppose that a given scaling algorithm produces a sequence (18) of diagonal equivalents for a nonnegative matrix* $\mathbf{A}$ *that satisfy the three properties (P1*), (P2) and (P3), given in Theorem 5 above, and if additionally*

1. *the algorithm is permutation insensitive, in the sense that, under any row or column permutation of the original matrix* $\mathbf{A}$*, the scaling elements remain the same and are just permuted along the diagonals of* $\mathbf{D}^{(k)}$ *and* $\mathbf{E}^{(k)}$ *with respect to the corresponding row or column permutations applied to* $\mathbf{A}$*,*

2. *the iterates also remain the same when applying the algorithm to a block diagonal matrix as when collecting results obtained by applying the algorithm on each diagonal submatrix separately,*

then, if $\mathbf{A}$ *has total support, both limits* $\mathbf{D} = \lim_{k\to\infty} \mathbf{D}^{(k)}$ *and* $\mathbf{E} = \lim_{k\to\infty} \mathbf{E}^{(k)}$ *do exist and* $\widehat{\mathbf{A}} = \mathbf{DAE}$ *is doubly stochastic.*

A.1 General considerations

We would like to conclude this part with some general comments with respect to the differences between the general convergence results detailed above and those raised by Parlett and Landis in [24]. As already recalled in the course of the previous discussion, Parlett and Landis, in their Theorem 1, have raised three main properties characterizing iterative scaling algorithms that would converge to a doubly stochastic matrix when the matrix A has support. Their first property is exactly the same as (P1) in Theorem 4. Property (P2) above brings in different simplifications with respect to the corresponding second property in [24]. Indeed, Parlett and Landis were supposing that, under the assumption that $\lim_{k\to\infty} \frac{s_k}{s_{k+1}} = 1$, the hypothesis in (P2) above hold as well as the fact that

$$\lim_{k\to\infty} d_i^{(k+1)}/d_i^{(k)} = \lim_{k\to\infty} e_j^{(k+1)}/e_j^{(k)} = 1, \;\; 1 \leq i,j \leq n \; .$$

As we can see from Lemma 1, the very first assumption is a direct consequence of (P1) and (P3) (which imply that the sequence $(s_k)_{k\geq 1}$ is convergent), and does not need to be verified in essence, and the two additional requirements just above are not necessary for the same results to hold in Theorem 4. For the third property, Parlett and Landis were in fact requiring that the algorithm incorporates a normalization step at each iteration, with

$$\mu_k = \frac{1}{n} \sum_{i=1}^{n} r_i^{(k+1)} = 1 \; ,$$

enforcing the mean of all row sums (or column sums, which are equal together) to be set to 1 at each iteration. Property (P3) above relaxes this equality, replacing it essentially just by an inequality.

The proof that each of the sequences of scaling factors $\left(d_i^{(k)}\right)_{k\geq 1}$ and $\left(e_j^{(k)}\right)_{k\geq 1}$ converge under the hypothesis that $\mathbf{A}$ has a total support is not contained in [24, Theorem 1]. Actually, the properties (P1), (P2), and (P3) above are not sufficient to reach to this conclusion. As we have shown, property (P1*), which is a strengthened form of the diagonal product increasing property (P1), with (P2) and (P3) together are sufficient for the conclusions in Theorem 5 to hold for any fully indecomposable nonnegative matrix, and the last two extra hypothesis in Corollary 2 enable to extend these conclusions to nonnegative matrices with total support only. These extra hypothesis can be thought as "*natural*" in general, in particular if the algorithm generates iterates that rely only on the computation of the one norm of both rows and columns at each iteration, as is the case for instance for both the Sinkhorn–Knopp algorithm and for our Algorithm 1 in section 3. A last comment about that concerns the difference between property (P3) and the corresponding one in [24]. Indeed, the requirement that the algorithm incorporates a normalization step, with the mean of all row sums set to 1 at each iteration, may however invalidate the second extra property required in Corollary 2 above, because the averaging of row sums may not be

the same when computed globally or block-wise in the case of a block-diagonal matrix. This may not prevent anyway the algorithms proposed in [24], that do respect this strengthened form of (P3), to produce still a convergent sequence of scaling factors. We have not found how to characterize in a simple way some common generic properties, that might enable to reach the same conclusions in the case of matrices with total support only, and extend the scope of Corollary 2 to address these particular "*normalized*" scaling algorithms.

The demonstration of the results in this appendix actually follows very closely the discussions and developments made by Parlett and Landis [24], and this improves only partially the already very general scope in their theoretical analysis. We have just included for completeness all the material about this generalisation in this appendix, since the extensions that these theorem are providing cover directly the specific case of our Algorithm 1 in the 1-norm, and we hope that this might be useful too in other circumstances to cover the convergence of scaling alternatives.

Bibliography

[1] Eva Achilles. Implications of convergence rates in Sinkhorn balancing. *Linear Algebra and its Applications*, 187:109–112, July 1993.

[2] P. R. Amestoy, I. S. Duff, D. Ruiz, and B. Uçar. A parallel matrix scaling algorithm. In J. M. Palma, P. R. Amestoy, M. Daydé, M. Mattoso, and J. C. Lopes, editors, *High Performance Computing for Computational Science - VECPAR 2008: 8th International Conference*, volume 5336 of *Lecture Notes in Computer Science*, pages 301–313. Springer Berlin / Heidelberg, 2008.

[3] Patrick R. Amestoy, Iain S. Duff, Jean-Yves L'Excellent, and Jacko Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.

[4] Patrick R. Amestoy, Abdou Guermouche, Jean-Yves L'Excellent, and Stéphane Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2):136–156, 2006.

[5] F. L. Bauer. Optimally scaled matrices. *Numerische Mathematik*, 5:73–87, 1963.

[6] F. L. Bauer. Remarks on optimally scaled matrices. *Numerische Mathematik*, 13:1–3, 1969.

[7] Abraham Berman and Robert J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. SIAM, 1994.

[8] G. D. Birkhoff. Tres observaciones sobre el algebra lineal. *Universidad Nacional de Tucuman Revista, Serie A*, 5:147–151, 1946.

[9] Alberto Borobia and Rafael Cantó. Matrix scaling: A geometric proof of Sinkhorn's theorem. *Linear Algebra and its Applications*, 268(1–3):1–8, January 1998.

[10] R. A. Brualdi, S. V. Parter, and H. Schneider. The diagonal equivalence of a nonnegative matrix to a stochastic matrix. *Journal of Mathematical Analysis and Applications*, 16:31–50, 1966.

[11] Richard A. Brualdi and Herbert J. Ryser. *Combinatorial Matrix Theory*, volume 39 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, UK; New York, USA; Melbourne, Australia, 1991.

[12] J. R. Bunch. Equilibration of symmetric matrices in the max-norm. *Journal of the ACM*, 18(4):566–572, 1971.

[13] J. R. Bunch and B. N. Parlett. Direct methods for solving symmetric indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 8(4):639–655, 1971.

[14] A. R. Curtis and J. K. Reid. On the automatic scaling of matrices for Gaussian elimination. *IMA Journal of Applied Mathematics*, 10(1):118–124, 1972.

[15] E. D. Dolan and J. J. More. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[16] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct Methods for Sparse Matrices*. Oxford University Press, London, 1986.

[17] I. S. Duff and S. Pralet. Strategies for scaling and pivoting for sparse symmetric indefinite problems. *SIAM Journal on Matrix Analysis and Applications*, 27(2):313–340, 2005.

[18] Iain S. Duff and Jacko Koster. The design and use of algorithms for permuting large entries to the diagonal of sparse matrices. *SIAM Journal on Matrix Analysis and Applications*, 20(4):889–901, 1999.

[19] Iain S. Duff and Jacko Koster. On algorithms for permuting large entries to the diagonal of a sparse matrix. *SIAM Journal on Matrix Analysis and Applications*, 22:973–996, 2001.

[20] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.

[21] Philip A. Knight. The Sinkhorn-Knopp algorithm: Convergence and applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1):261–275, 2008.

[22] Philip A. Knight and Daniel Ruiz. A fast algorithm for matrix balancing. *IMA Journal of Numerical Analysis*, 2012.

[23] Oren E. Livne and Gene H. Golub. Scaling by binormalization. *Numerical Algorithms*, 35:97–120, 2004.

[24] B. N. Parlett and T. L. Landis. Methods for scaling to double stochastic form. *Linear Algebra and its Applications*, 48:53–79, 1982.

[25] H. Perfect and L. Mirsky. The distribution of positive elements in doubly stochastic matrices. *The Journal of the London Mathematical Society*, 40:689–698, 1965.

[26] U. G. Rothblum, H. Schneider, and M. H. Schneider. Scaling matrices to prescribed row and column maxima. *SIAM Journal on Matrix Analysis and Applications*, 15(1):1–14, 1994.

[27] D. Ruiz. A scaling algorithm to equilibrate both rows and columns norms in matrices. Technical Report RAL-TR-2001-034 and RT/APO/01/4, Rutherford Appleton Laboratory, Oxon, UK and ENSEEIHT-IRIT, Toulouse, France, 2001.

[28] M. H. Schneider and S. Zenios. A comparative study of algorithms for matrix balancing. *Operations Research*, 38(3):439–455, 1990.

[29] R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35:876–879, 1964.

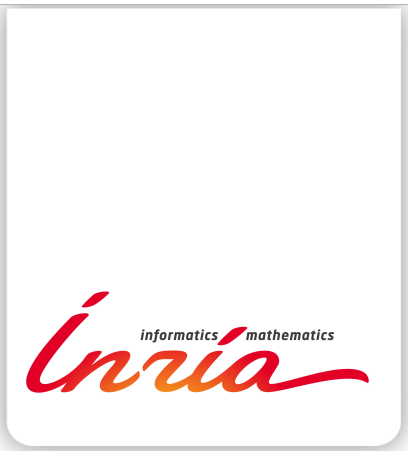
[30] R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *American Mathematical Monthly*, 74:402–405, 1967.

[31] R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[32] R. Sinkhorn and P. Knopp. Problems concerning diagonal products in nonnegative matrices. *Transactions of the American Mathematical Society*, 136:67–75, 1969.

[33] George W. Soules. The rate of convergence of Sinkhorn balancing. *Linear Algebra and its Applications*, 150:3–40, 1991. Proceedings of the First Conference of the International Linear Algebra Society (Provo, UT, 1989).

[34] A. van der Sluis. Condition numbers and equilibration of matrices. *Numerische Mathematik*, 14:14–23, 1969.



RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399